Power BI Architecture Lab Guide

Edition: November 2024

Author: Reza Rad

RADACAD Systems Limited

Contact Information: training@radacad.com

Reproduction or distribution of RADACAD materials in whole or in part is prohibited except by written permission. Audio or video recording of RADACAD classes is also prohibited.

Contents

Prerequisite 3
Lab exercise 1: Multi-layered Architecture 5
Lab Exercise 2: Powershell Cmdlets for Power BI 44

Prerequisite

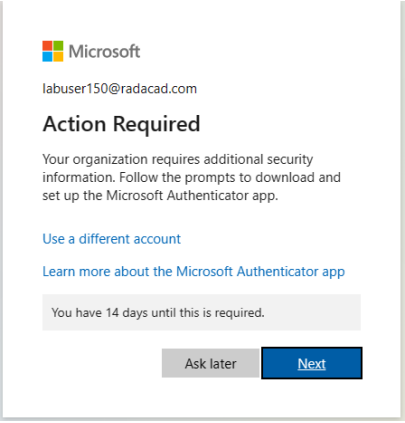
Download the lab files on your local folder and unzip it.

You need to have a Power BI Pro, PPU, or Premium/Microsoft Fabric capacity license and user. If you don't have these, please use the lab user created for you for this course. (Please ask your instructor about it)

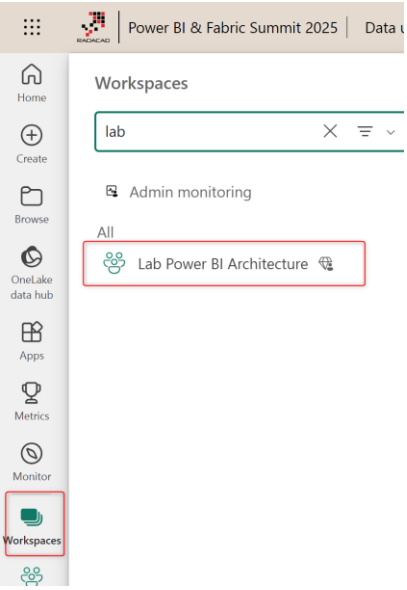
If you have any questions, please reach out to us at training@radacad.com

To use RADACAD's lab user accounts, follow the below instructions;

1. Open url: https://app.powerbi.com/ or https://app.fabric.microsoft.com/
2. Enter your Office365 user and password
3. If you are prompted to set up multi-factor authentication, select "**Ask Later**"



4. Once logged in, check that you see the below workspace; (If you don't see it, refresh the browser and try again)

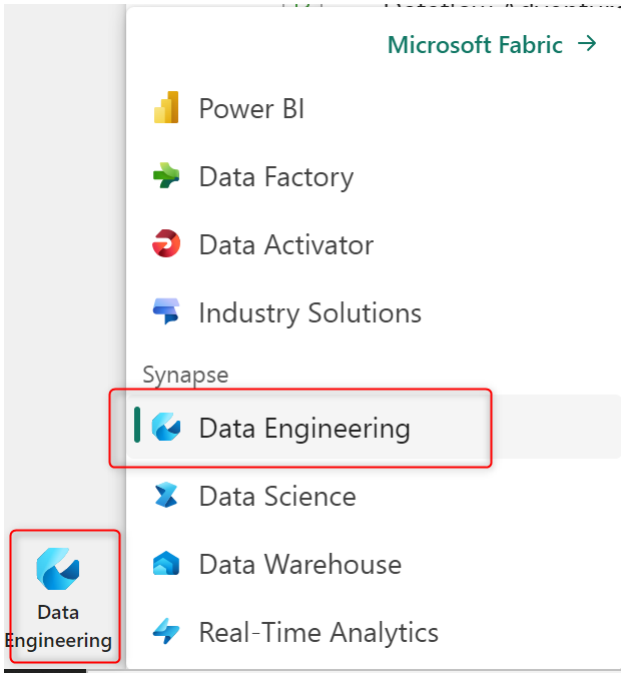


5. Select the workspace, and create a folder there with your name. for the rest of this lab, create all your items under that workspace and that folder. Also create every new item with your name in it as some of the items don't automatically get created under folders.

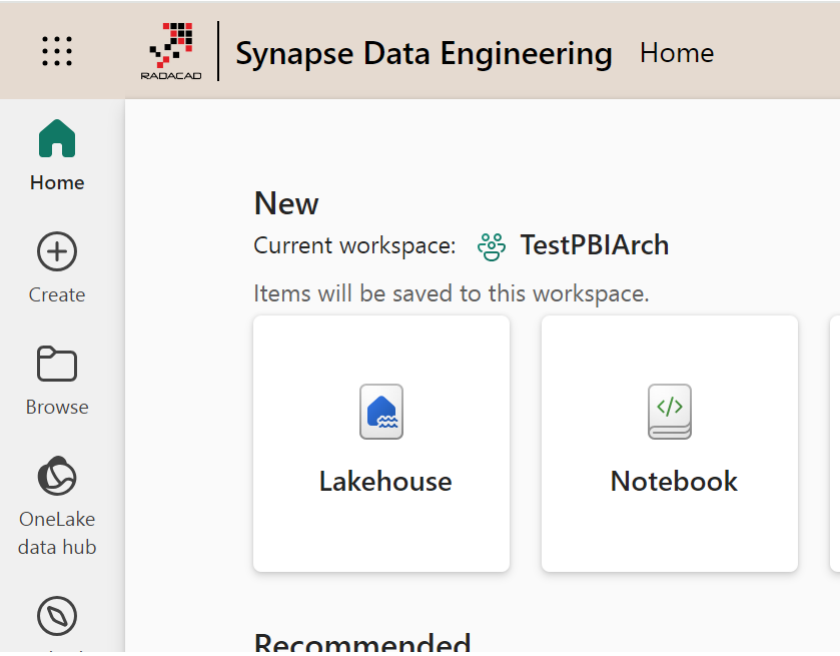
Lab exercise 1: Multi-layered Architecture

Stage 1: Creating General tables

1. Go to the Data Engineering workload

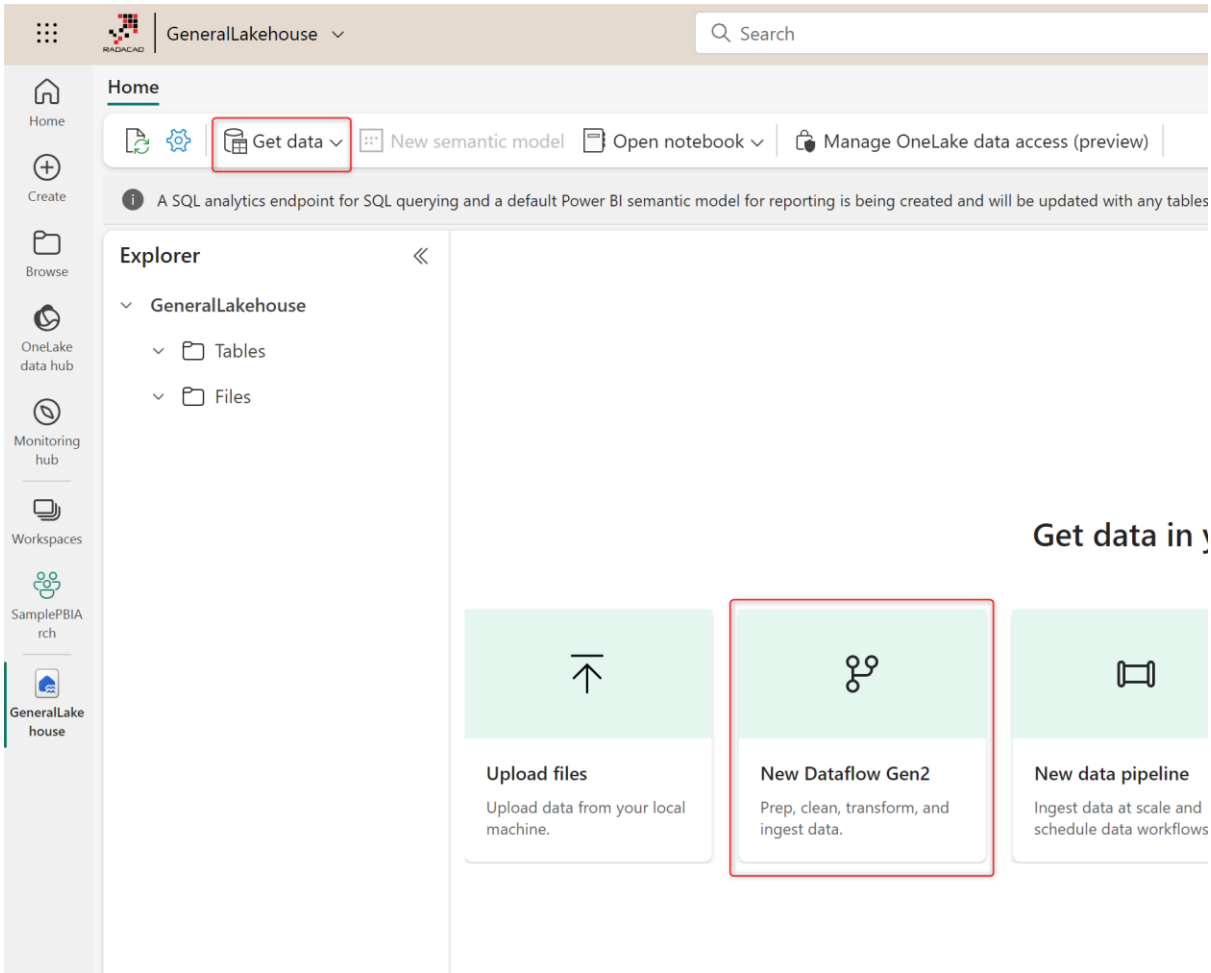


2. Create a Lakehouse and name it GeneralLakehouse



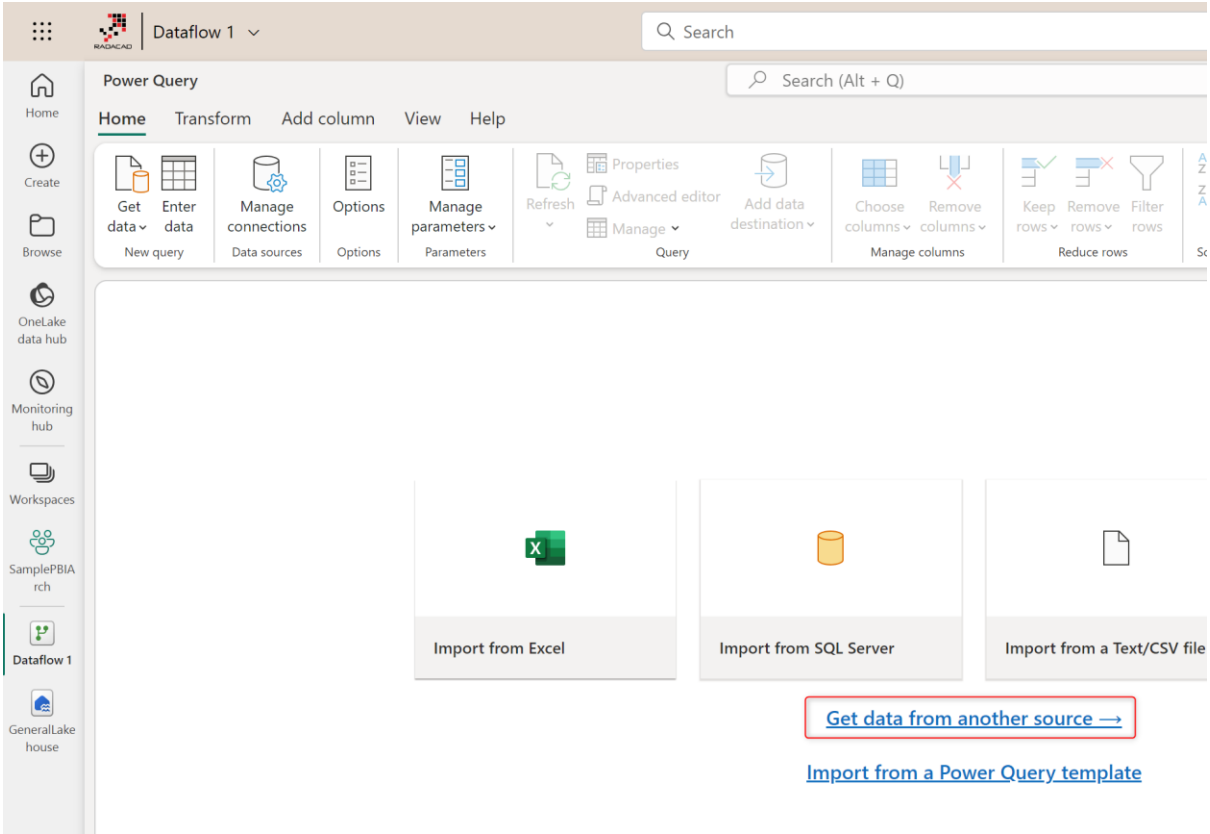
3. In the Lakehouse Explorer, click on Get Data and Choose Dataflow Gen2

Training@radacad.com

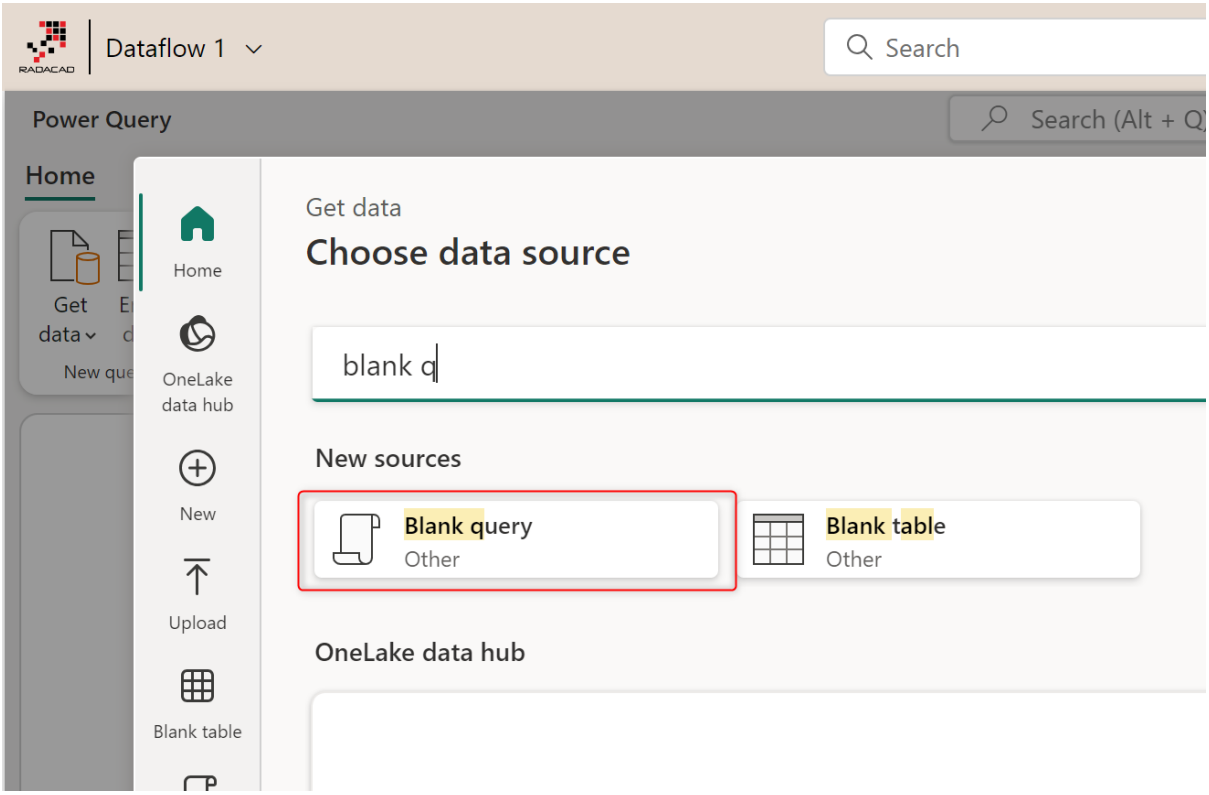


4. In Dataflow Gen2 Editor, Get data from another source

Training@radacad.com

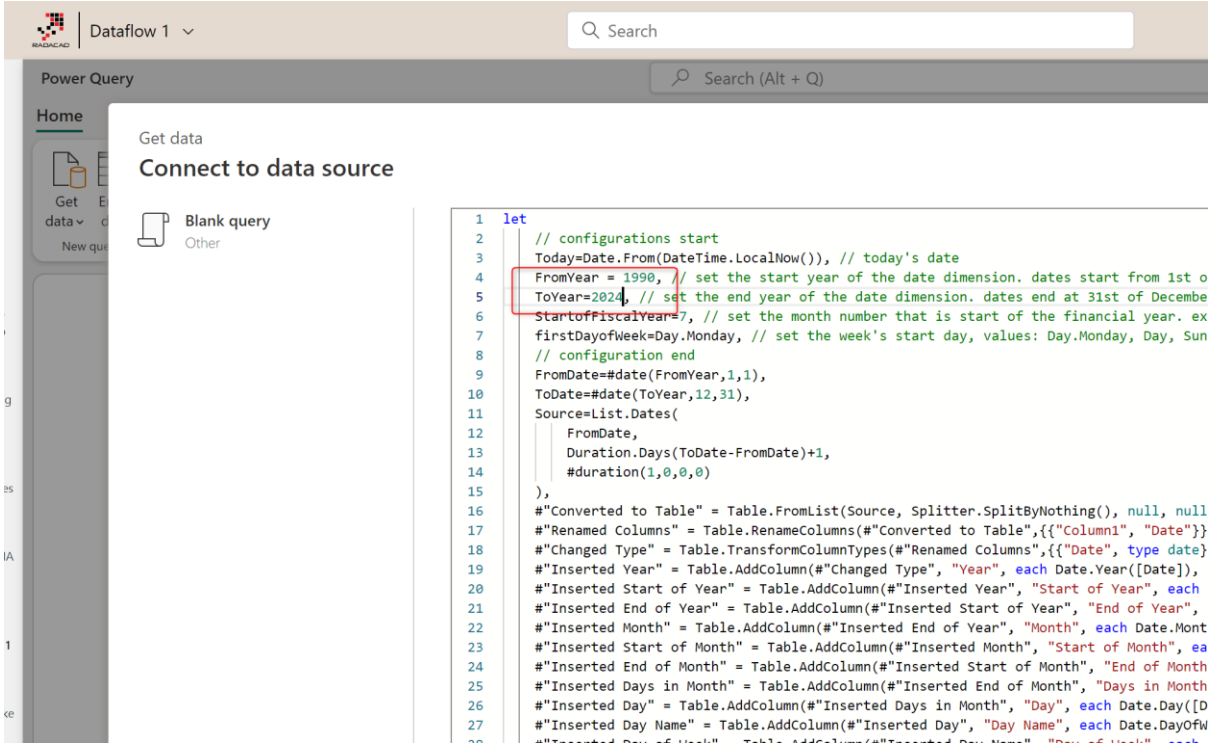


5. and then choose Blank Query as the source;

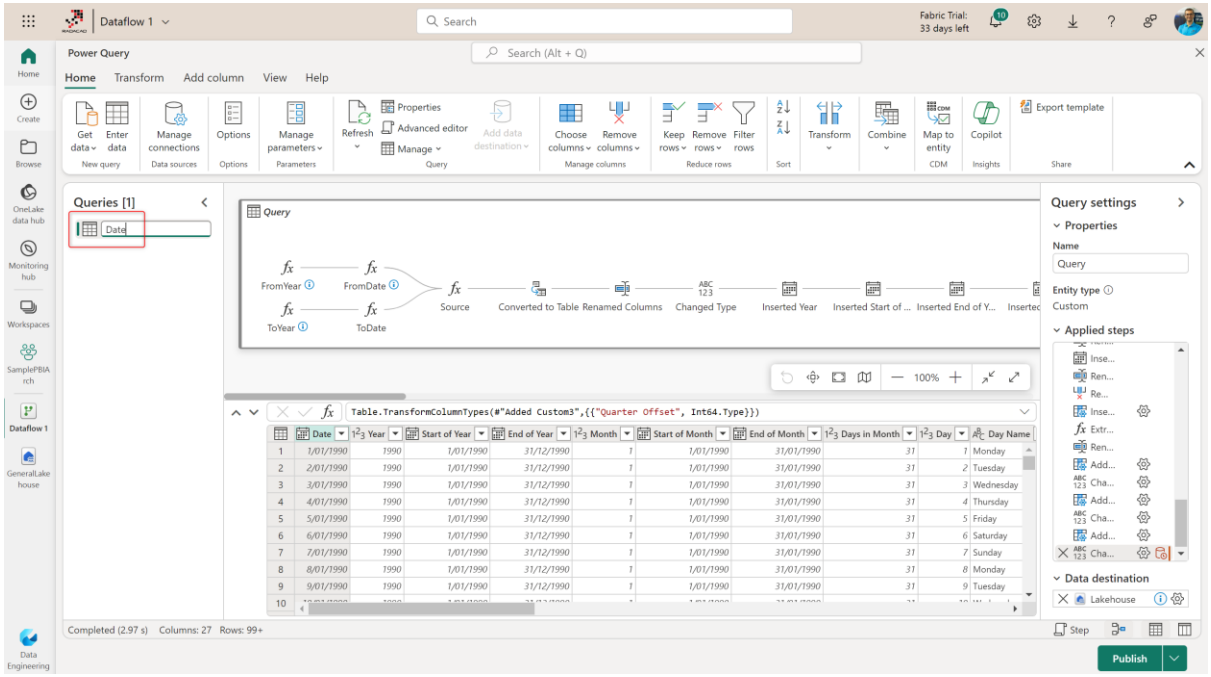


Training@radacad.com

6. Copy and paste the M script from the files (script-for-date-dimension.txt file) for this guide into here, change the start year to 1990 and end year to the current year. Then click on OK

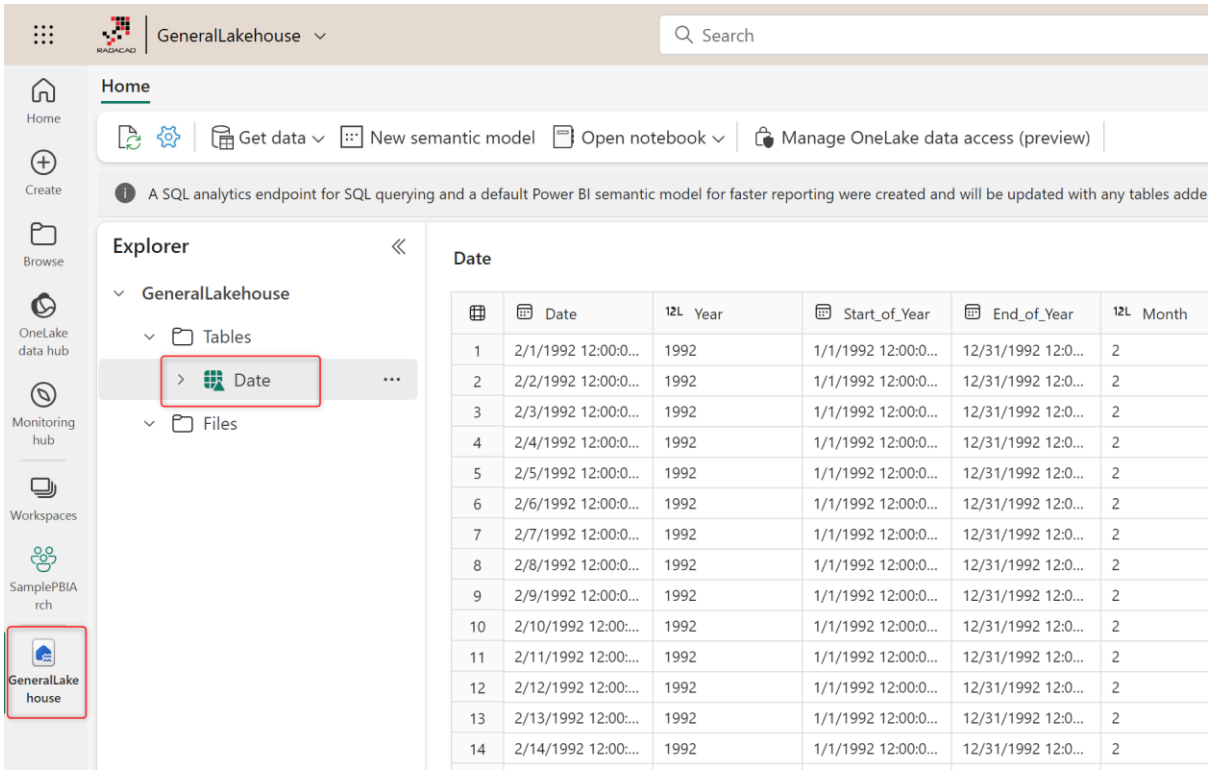


7. Once the table is generated, rename it to Date. And then Publish. (You can set the name for Dataflow before publishing it, to something like General Dataflow)



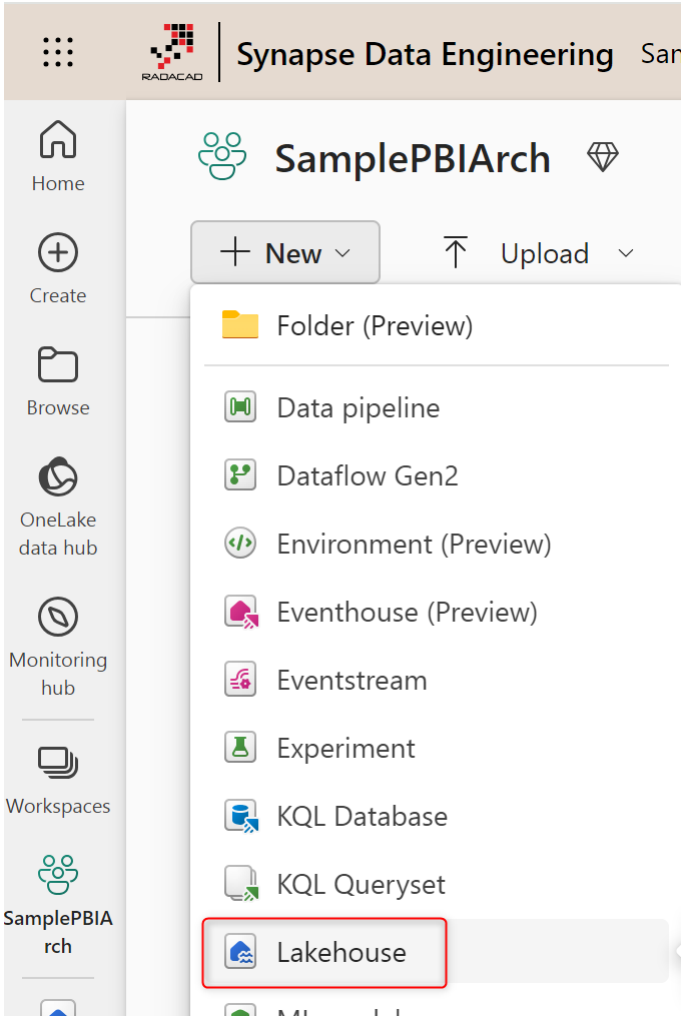
8. Wait for the refresh to finish, and then click on Lakehouse and you can see the Date table in the Lakehouse explorer.

Training@radacad.com



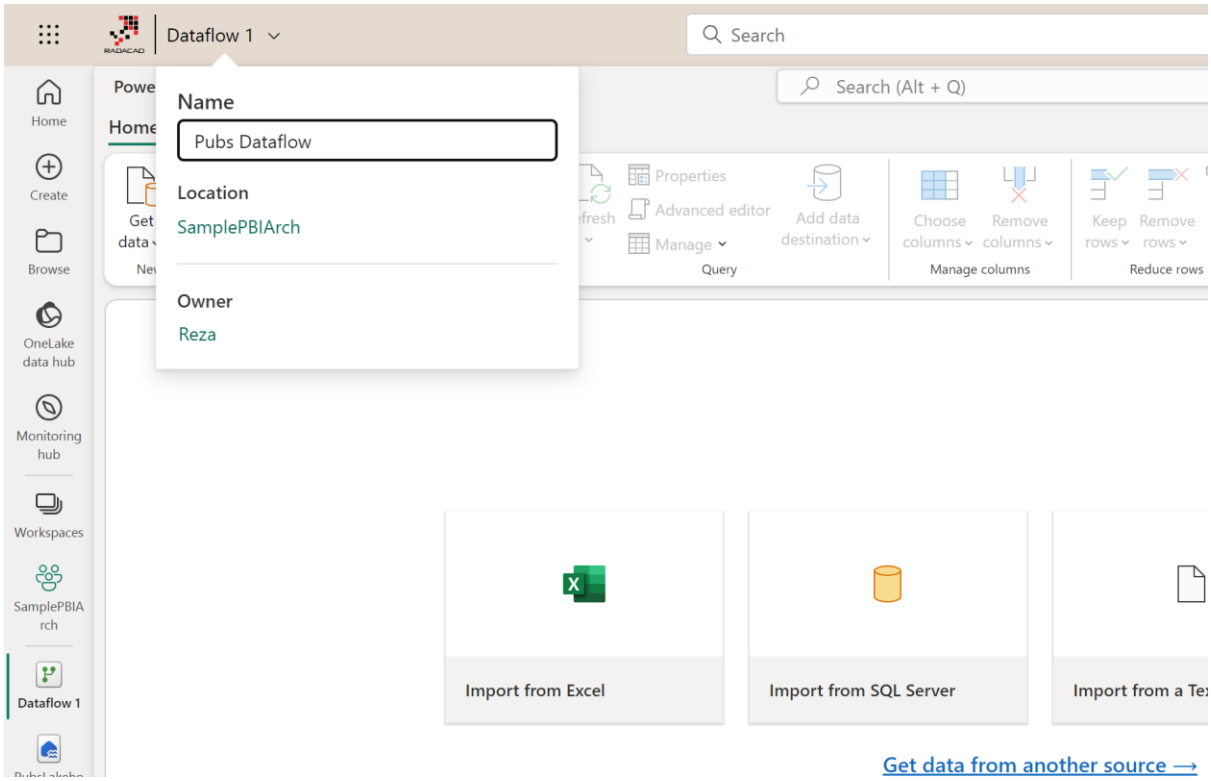
Stage 2: Creating Pubs solution

9. Create a new Lakehouse and name it, PubsLakehouse

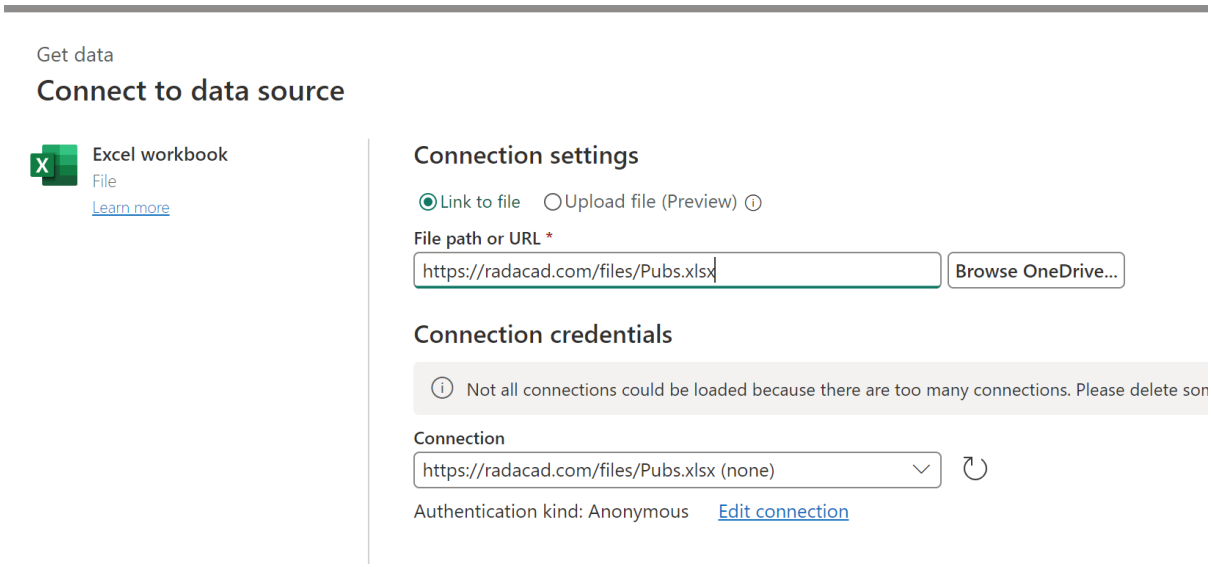


10. In the Lakehouse explorer for the PubsLakehouse, go to Get Data and choose Dataflow Gen2. Name this new Dataflow as Pubs Dataflow.

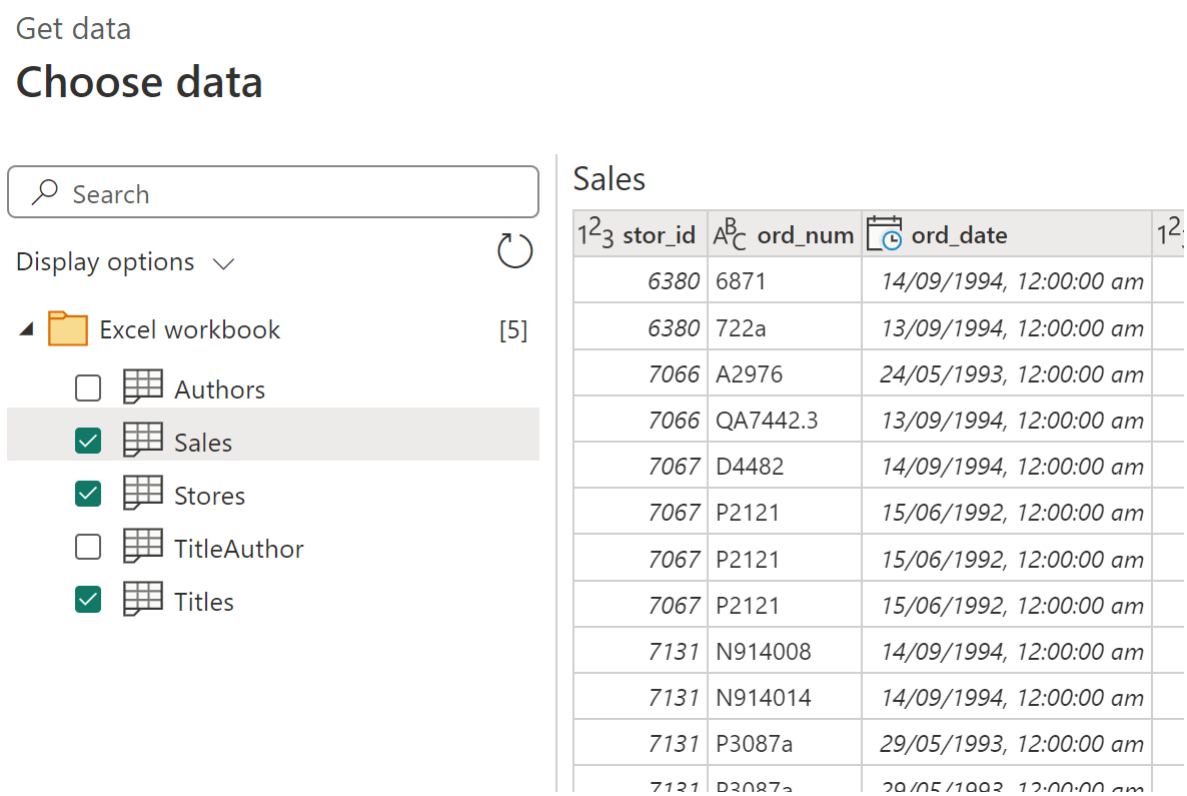
Training@radacad.com



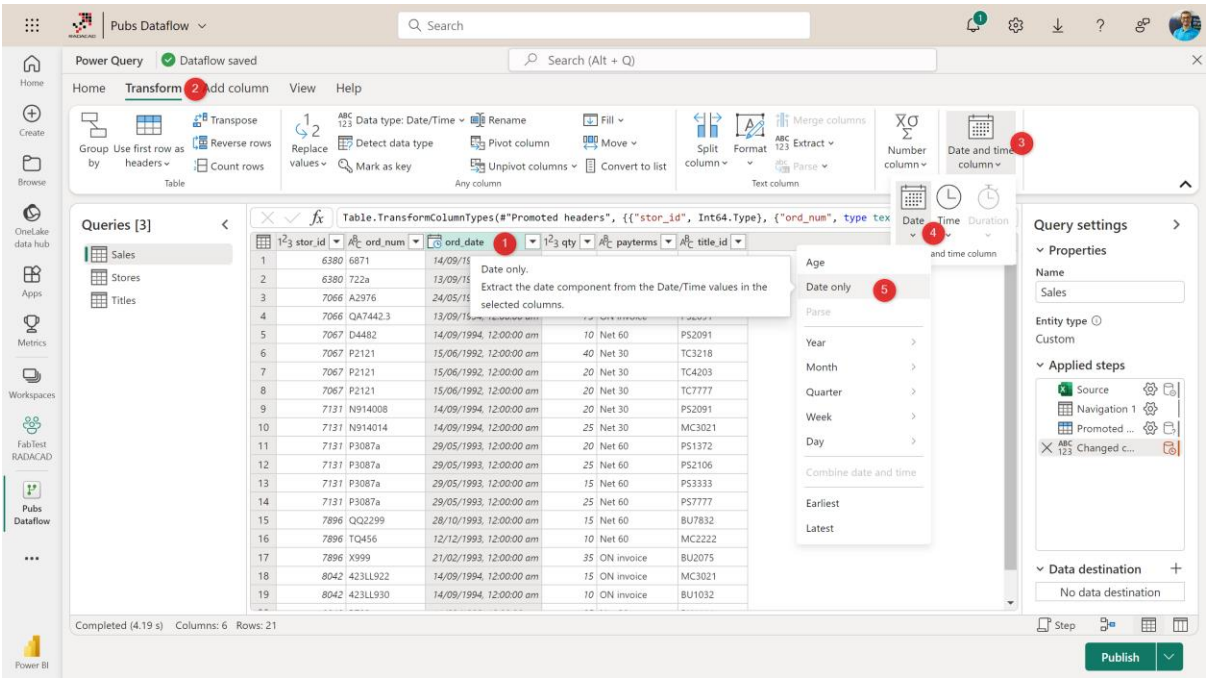
11. Choose Get data from Excel, and enter this URL as the source for Excel file: https://radacad.com/files/Pubs.xlsx



12. Select the tables Sales, Titles and Stores, and then go to the next step.

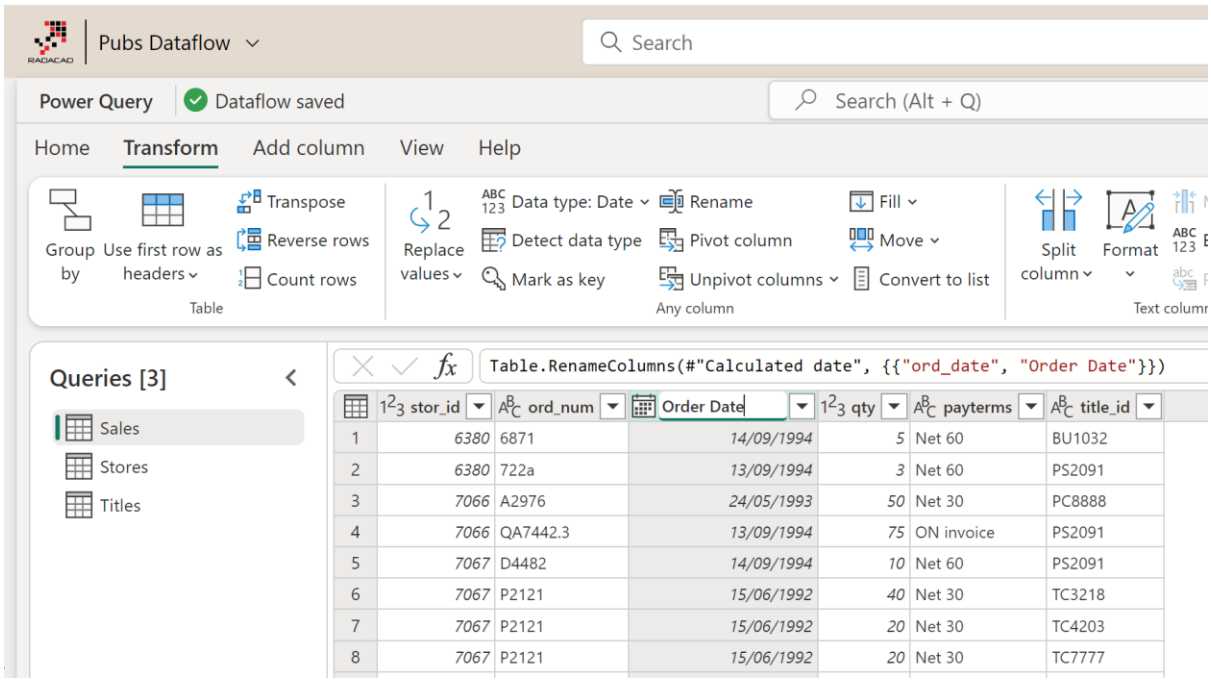


13. Change the ord_date field in the Sales table to be Date Only

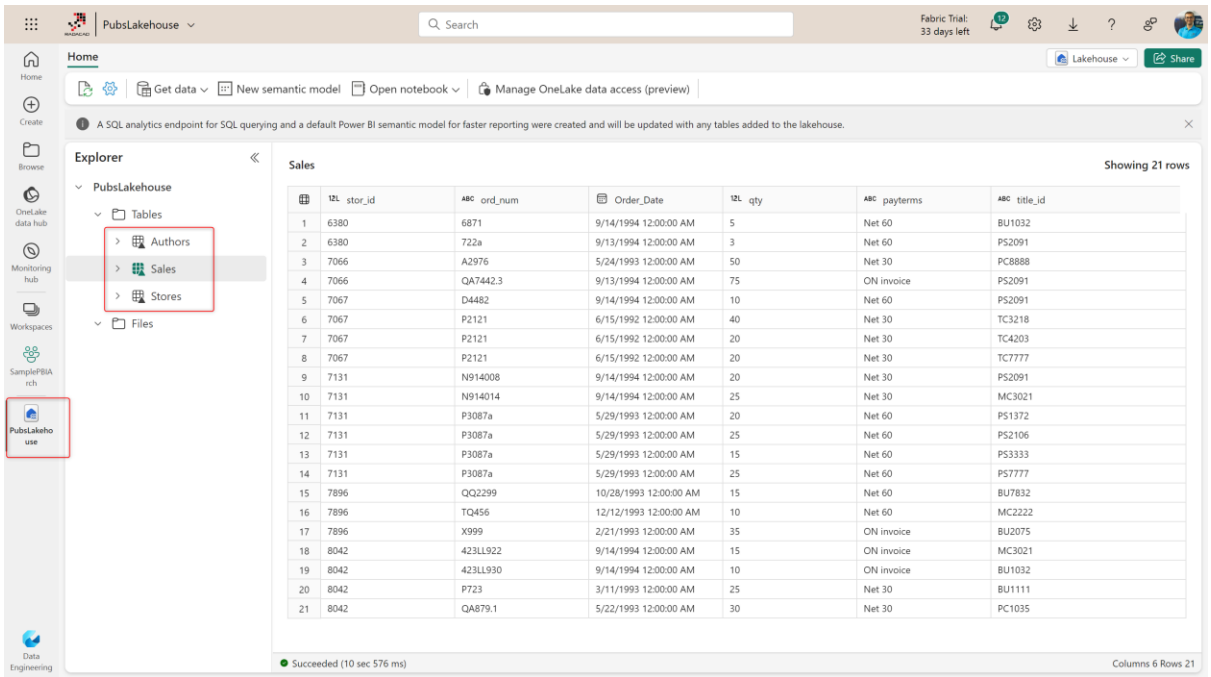


14. Rename this field to Order Date, and then Publish.

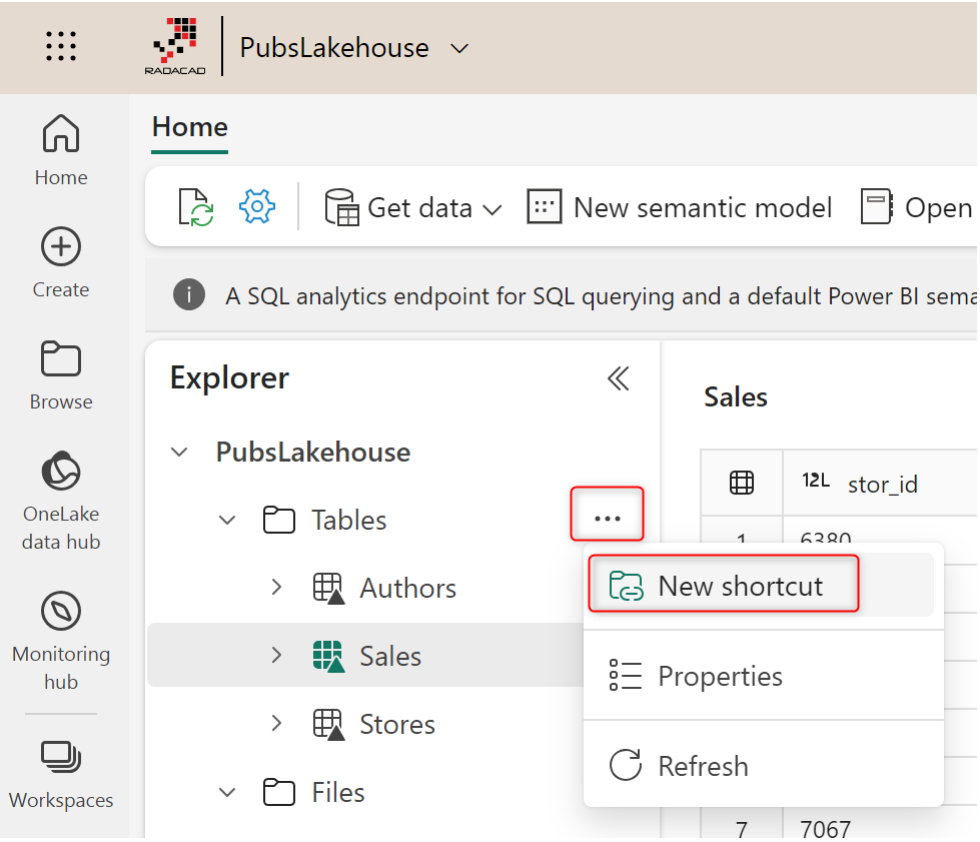
Training@radacad.com



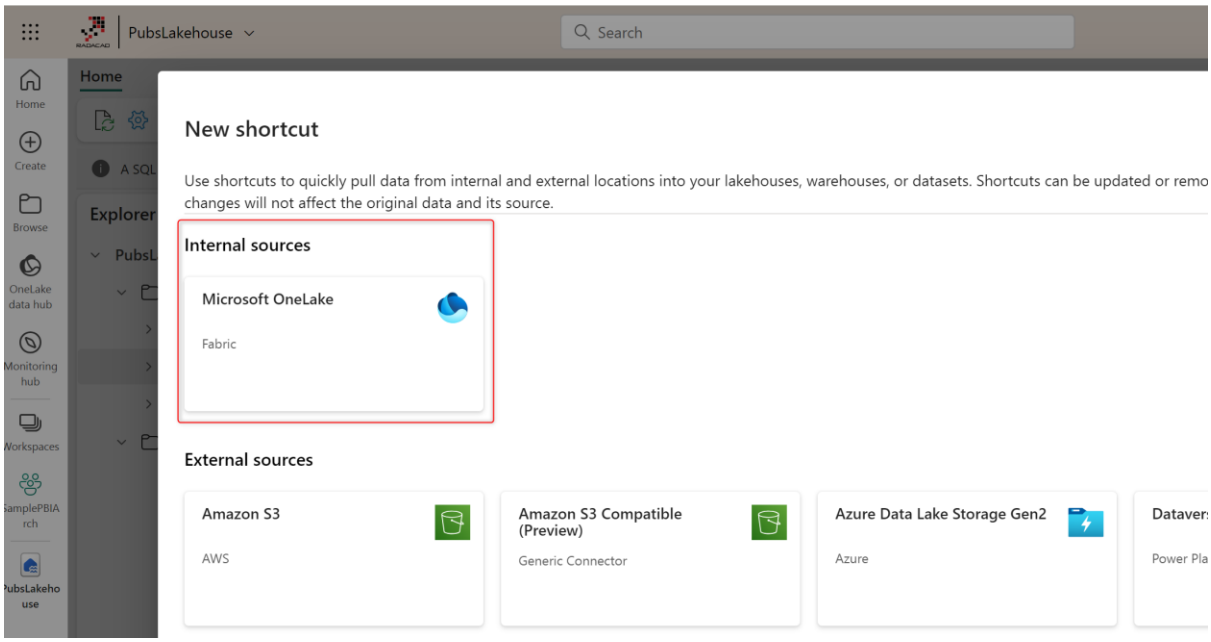
15. Wait for the refresh of the Dataflow to finish after the publish. Once finished, go to the Lakehouse explorer for the PubsLakehouse and check for existence of tables in there.



16. On the tables of the Lakehouse Explorer, click on more options (…) and select a New Shortcut

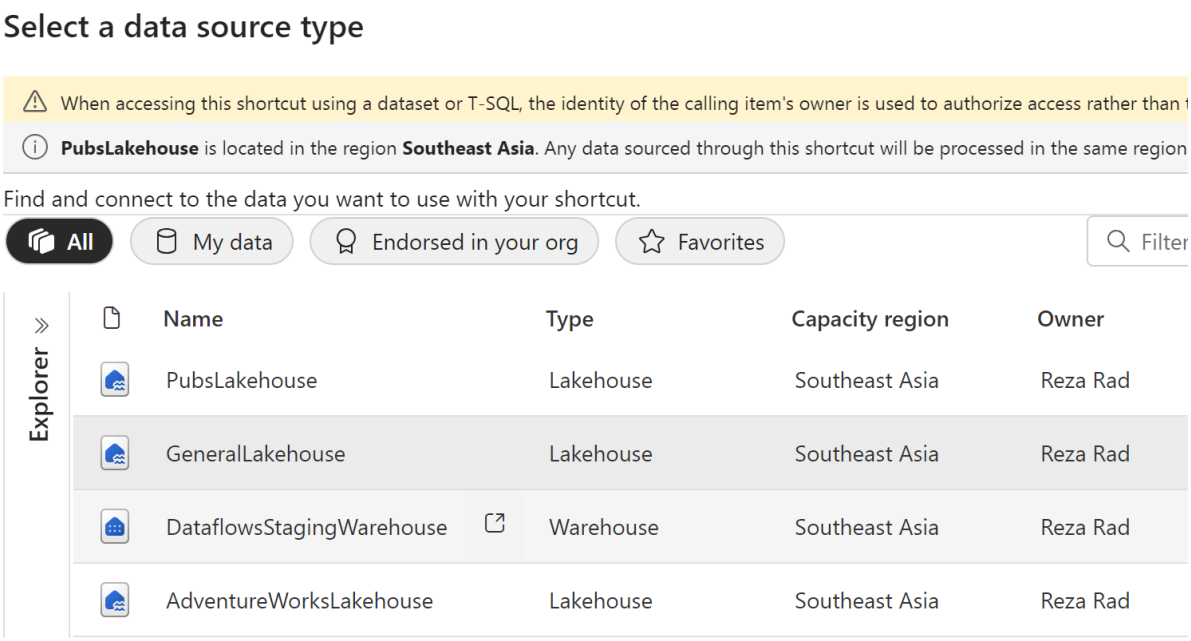


17. Select the Microsoft OneLake as the source of Shortcut;

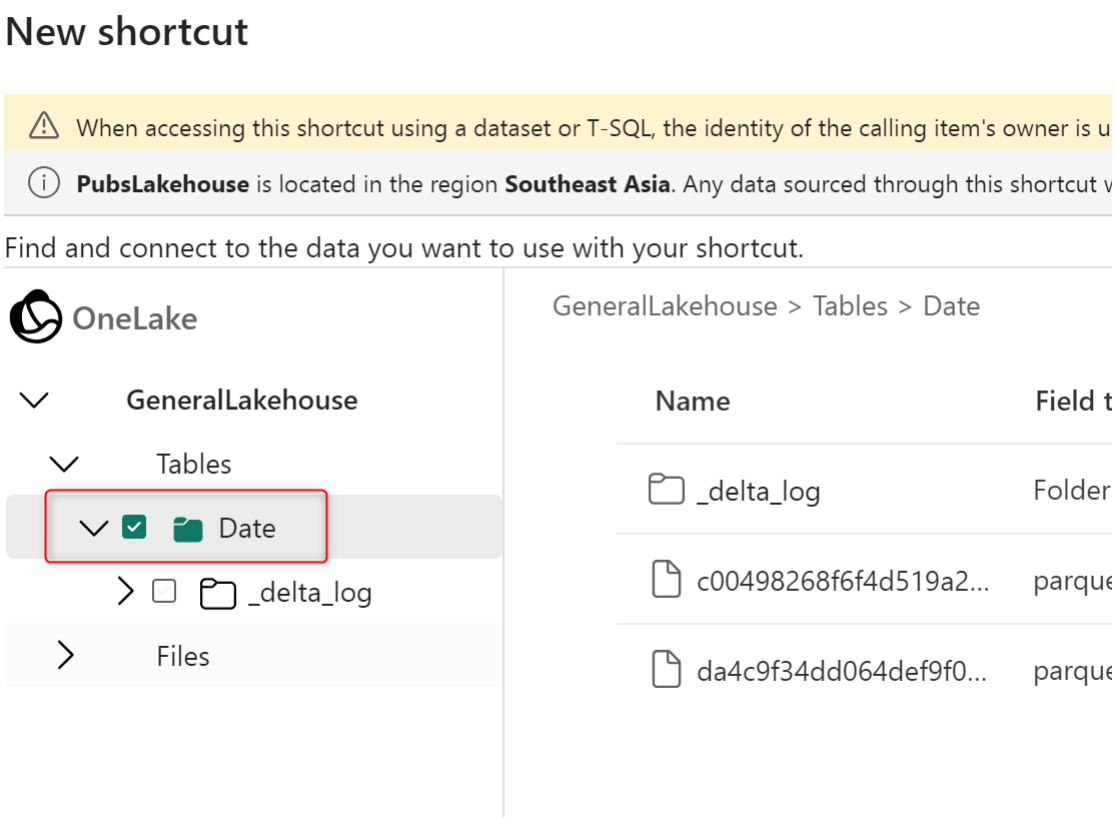


18. Choose the GeneralLakehouse

Training@radacad.com

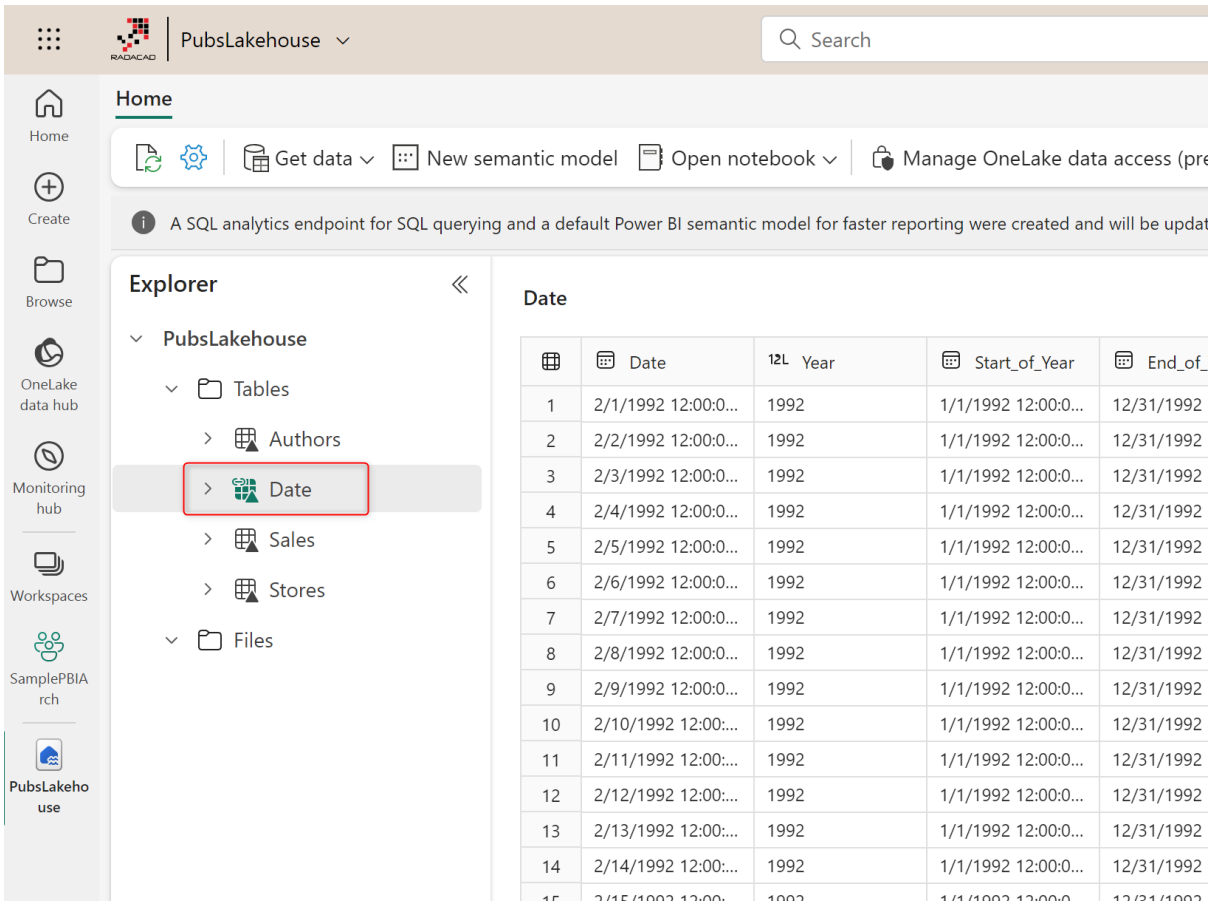


19. And under the tables, select Date.



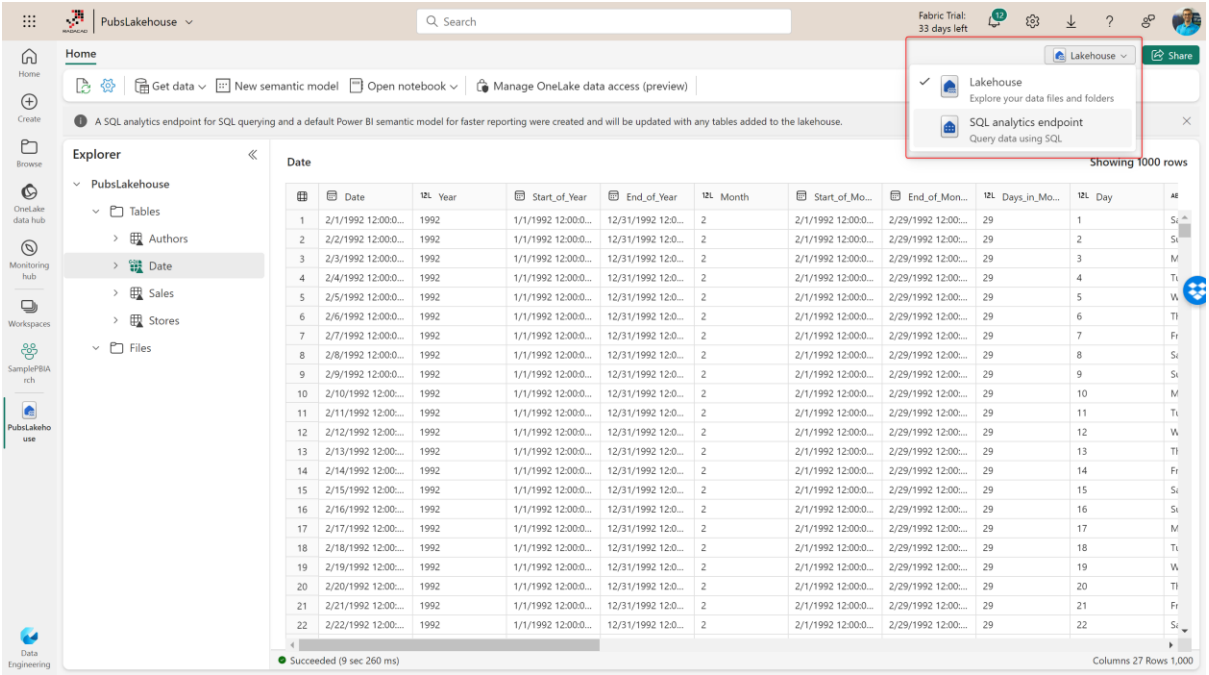
Training@radacad.com

20. Once clicked on Next and created the Shortcut, the new Shortcut table will appear under the tables as a normal table.

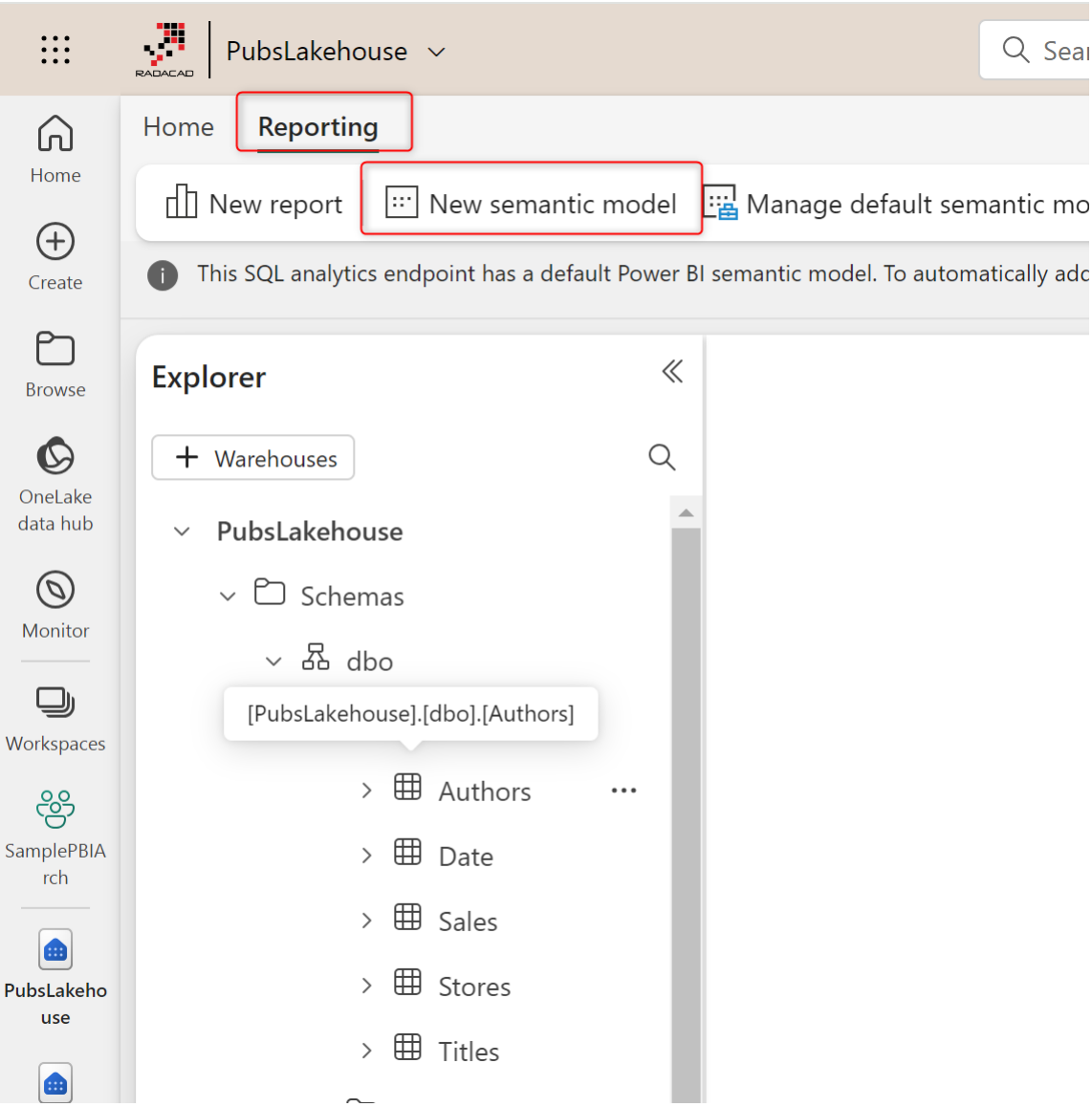


The shortcut means that the Date table's data remains in the GeneralLakehouse, but it will be accessible from this PubsLakehouse as a link too.

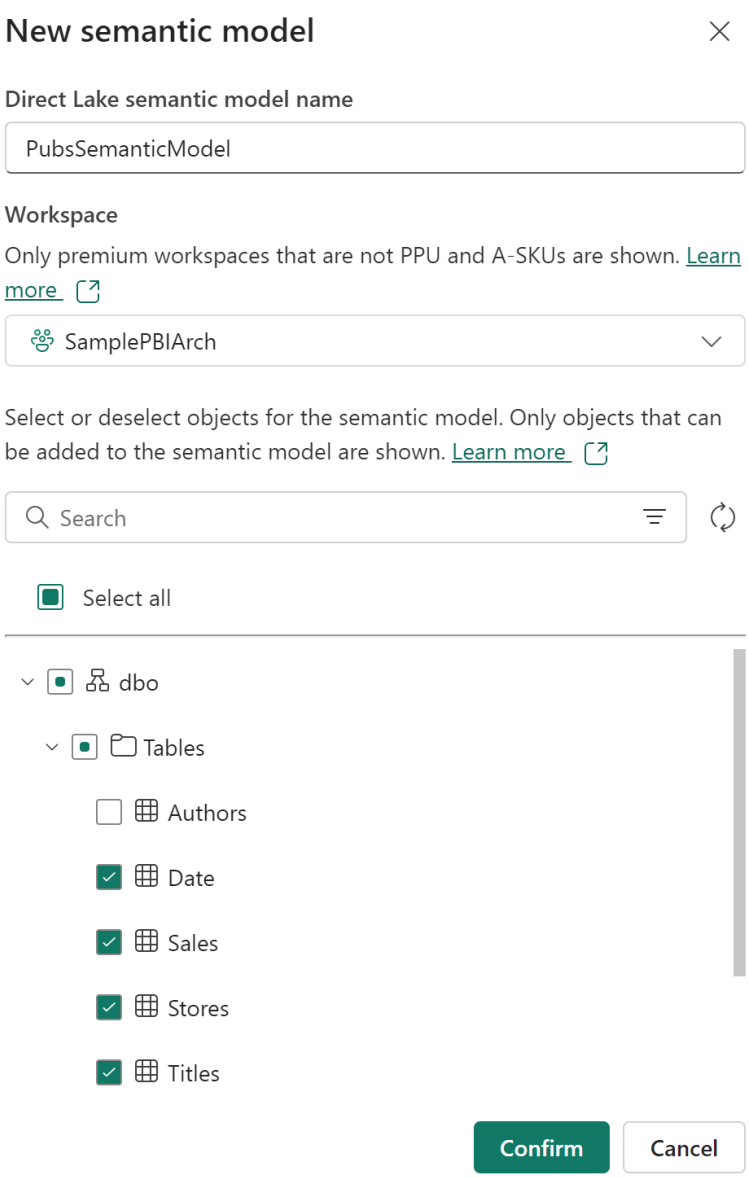
21. Navigated to the SQL Analytics Endpoint for the PubsLakehouse.



22. Note that all the four tables looks the same in this view, even the Date table is like any other tables. Navigate to the Reporting tab and create a New semantic Model.



23. Name this model as PubsSemanticModel and add all tables to it.

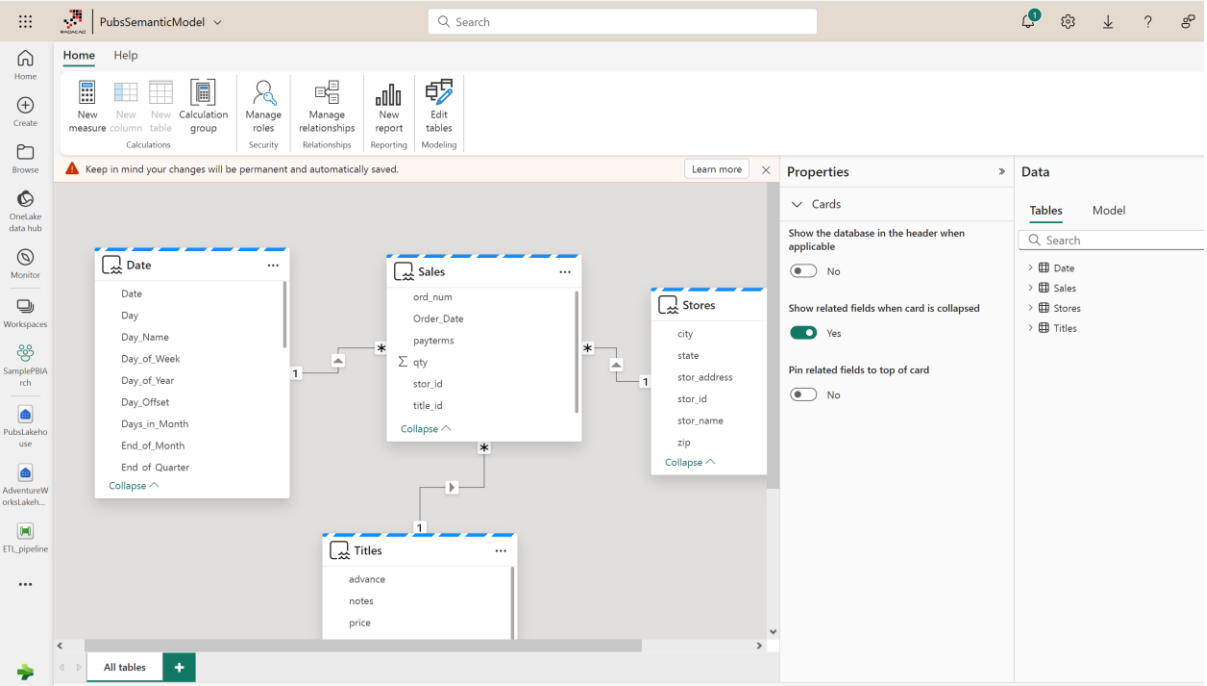


24. When the new model editor opens, Design tables with the relationships like the below screenshot;

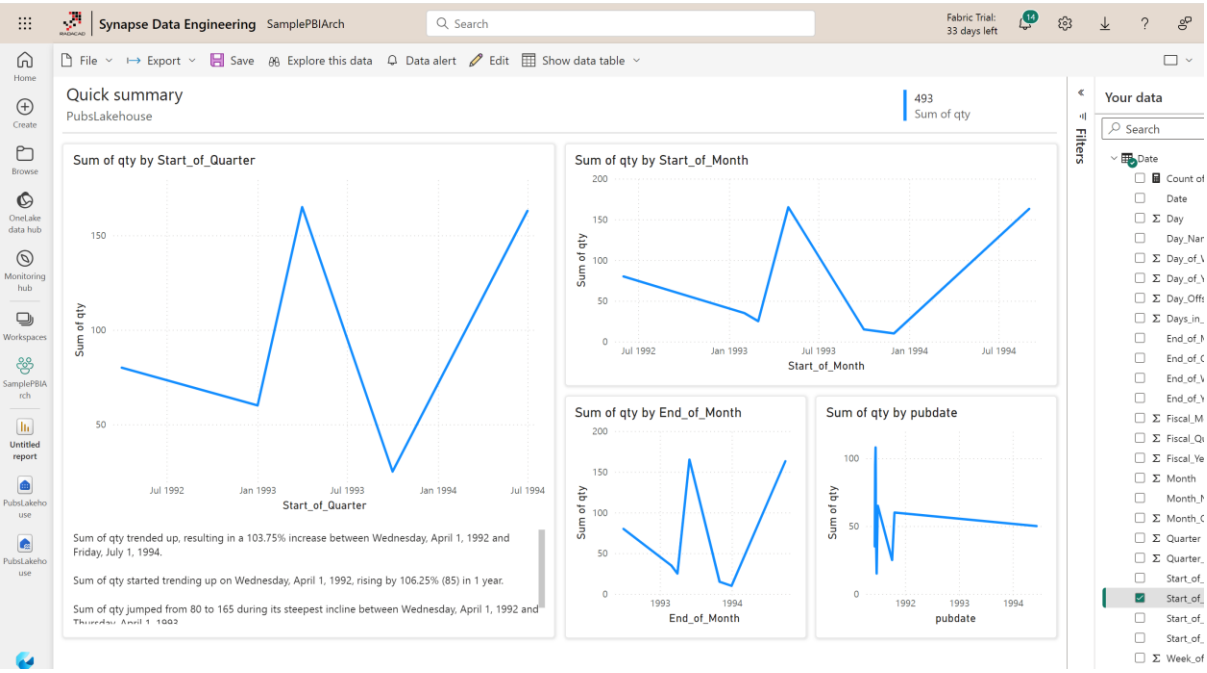
Date (Date) to Sales (Order Date)

Titles (title_id) to Sales (title_id)

Stores (stor_id) to Sales (stor_id)



25. Once completed, Create a new report from this. Below is a sample report generated; (yours doesn't have to be the same)



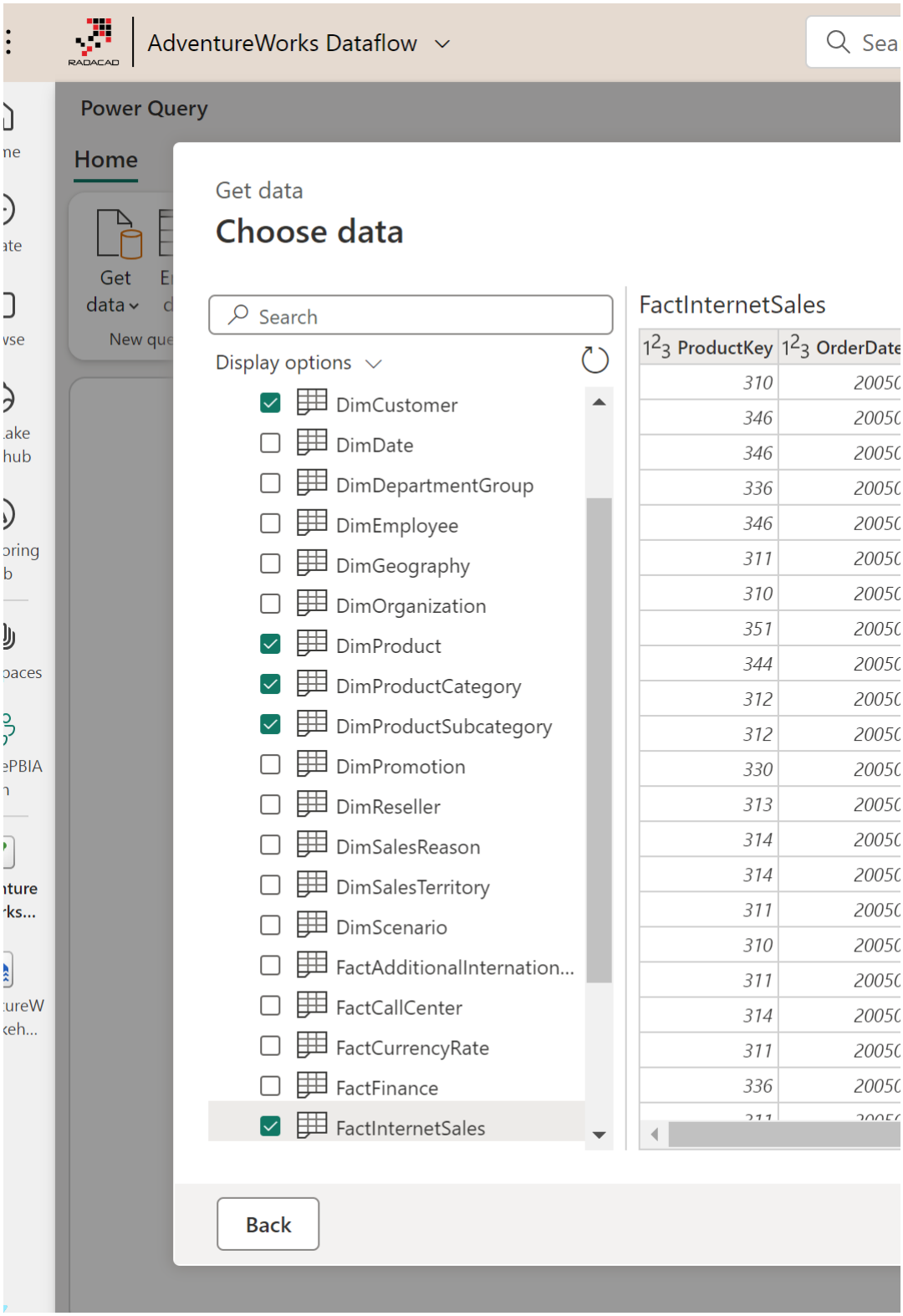
26. Save the report in the workspace.

Stage 3: Creating the AdventureWorks Solution

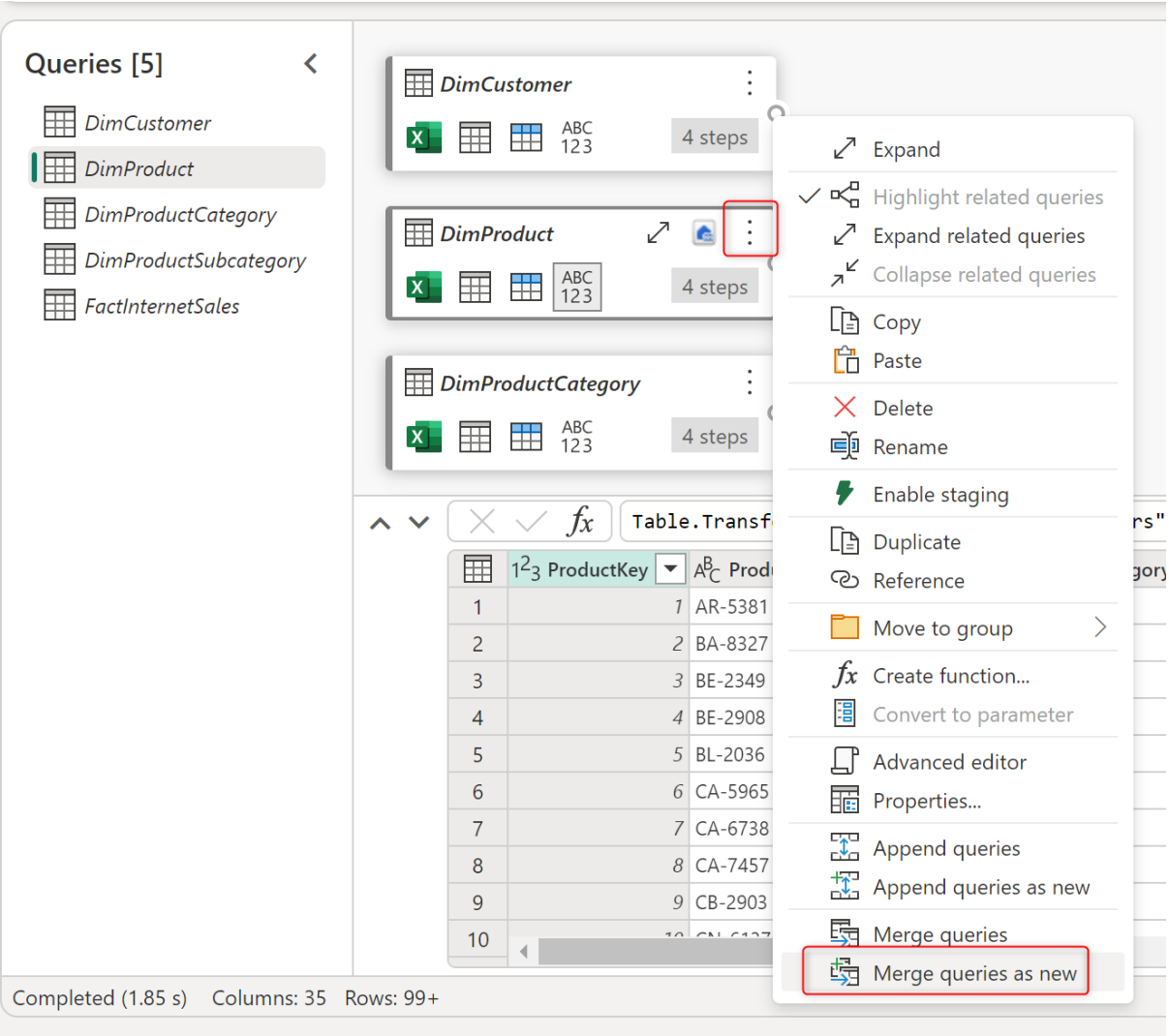
27. Create a new Lakehouse and name it AdventureWorksLakehouse

Training@radacad.com

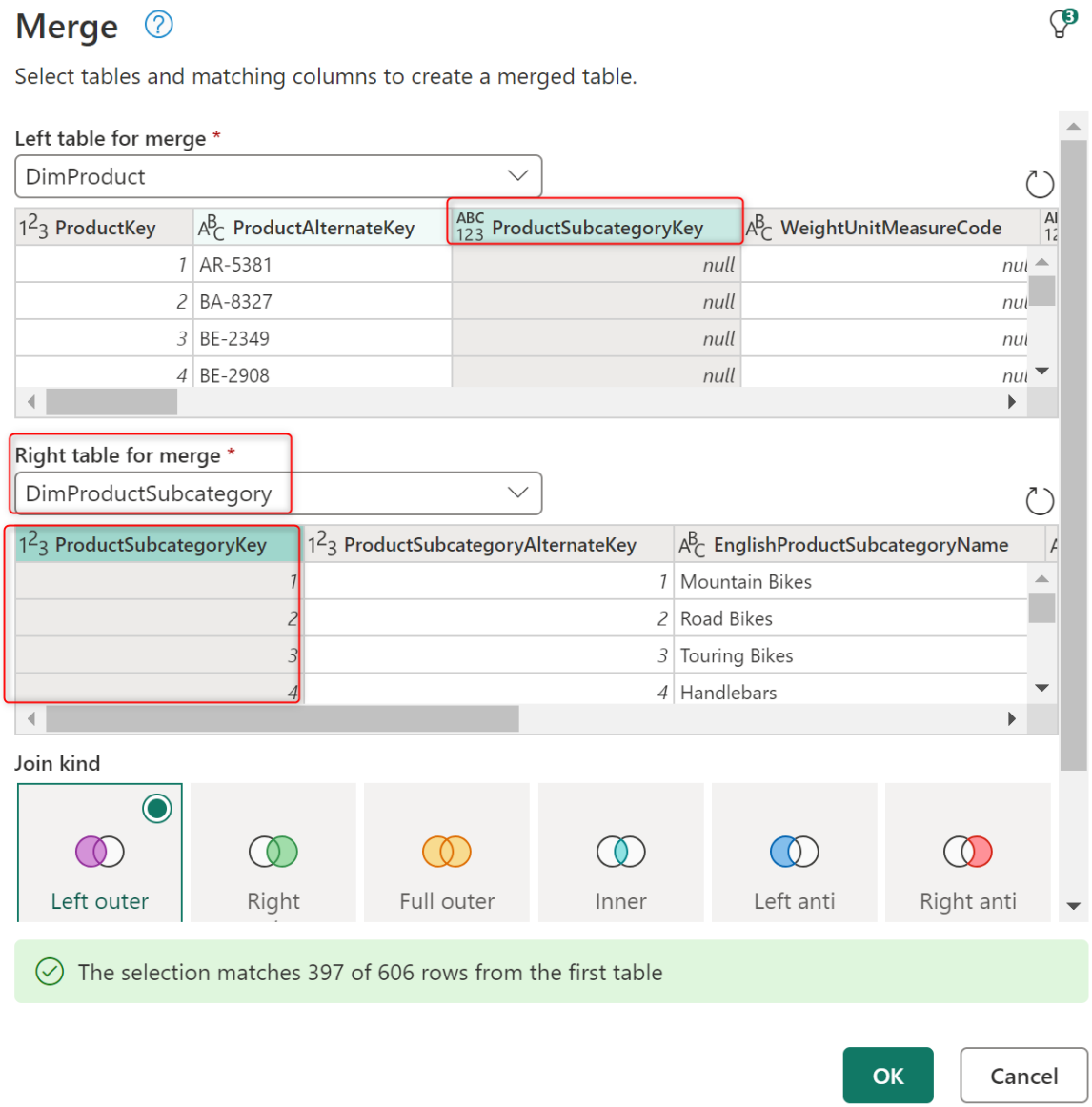
28. Then use Get Data using Dataflow Gen2, Name the Dataflow as AdventureWorks Dataflow, get data from Excel, and use this URL as the source: https://radacad.com/files/AdventureWorksDW2012.xlsx
29. Select the tables: DimCustomer, DimProduct, DimProductCategory, DimProductSubcategory, and FactInternetSales, and proceed to the next step.



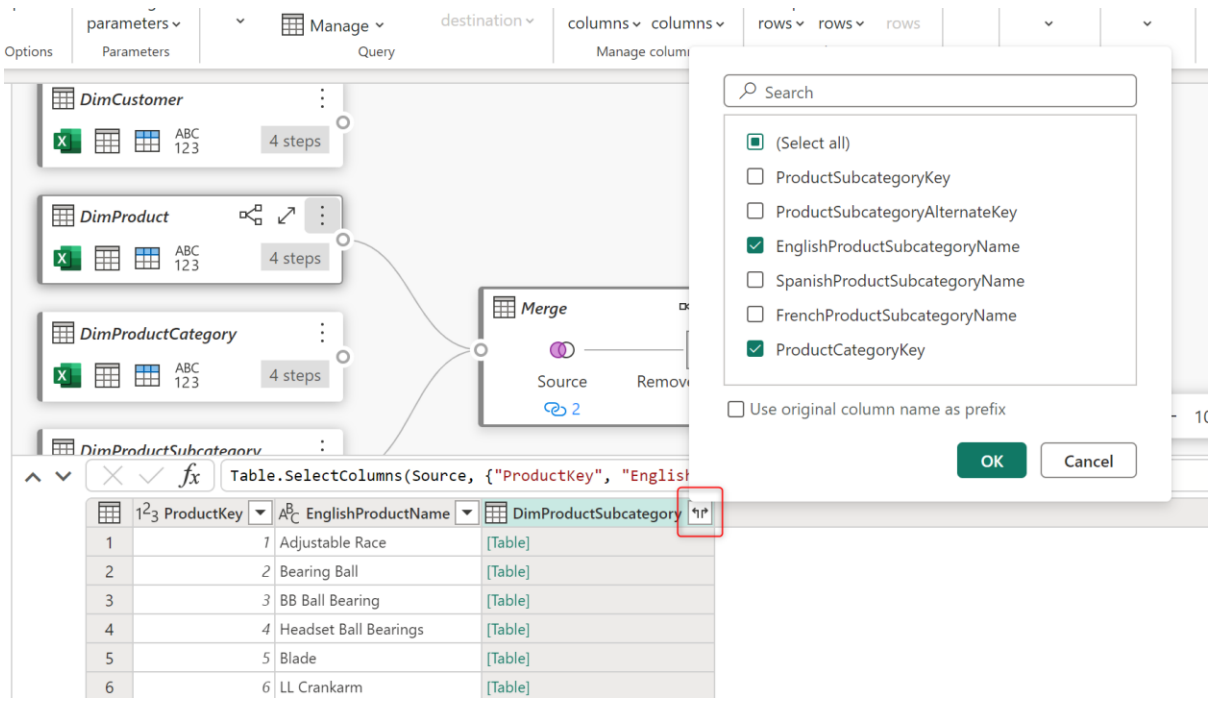
30. From the DimProduct table, Choose Merge Queries as New



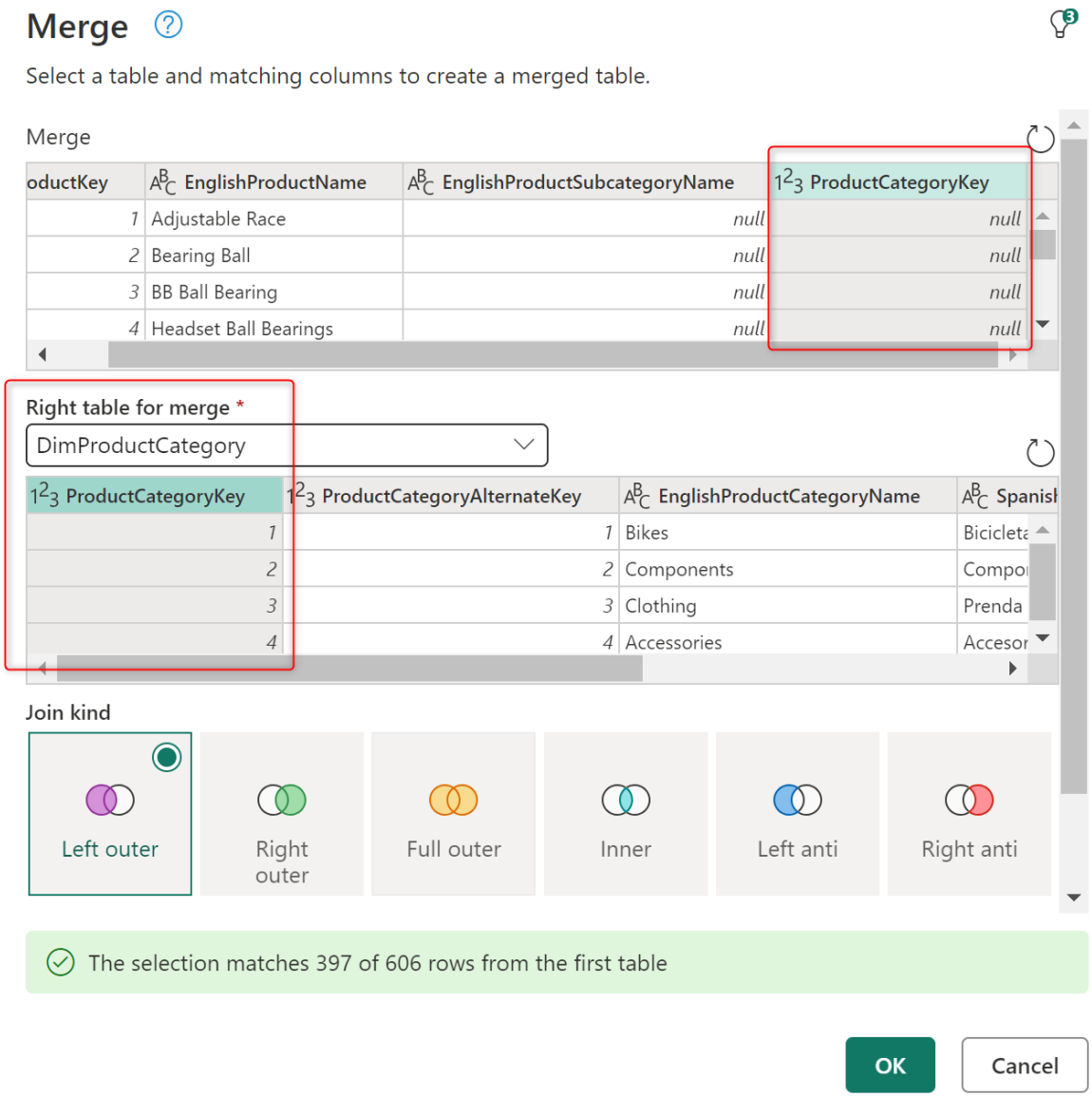
31. Choose the column to join as ProductSubcategoryKey, and the table as DimProductSubcategory and continue.



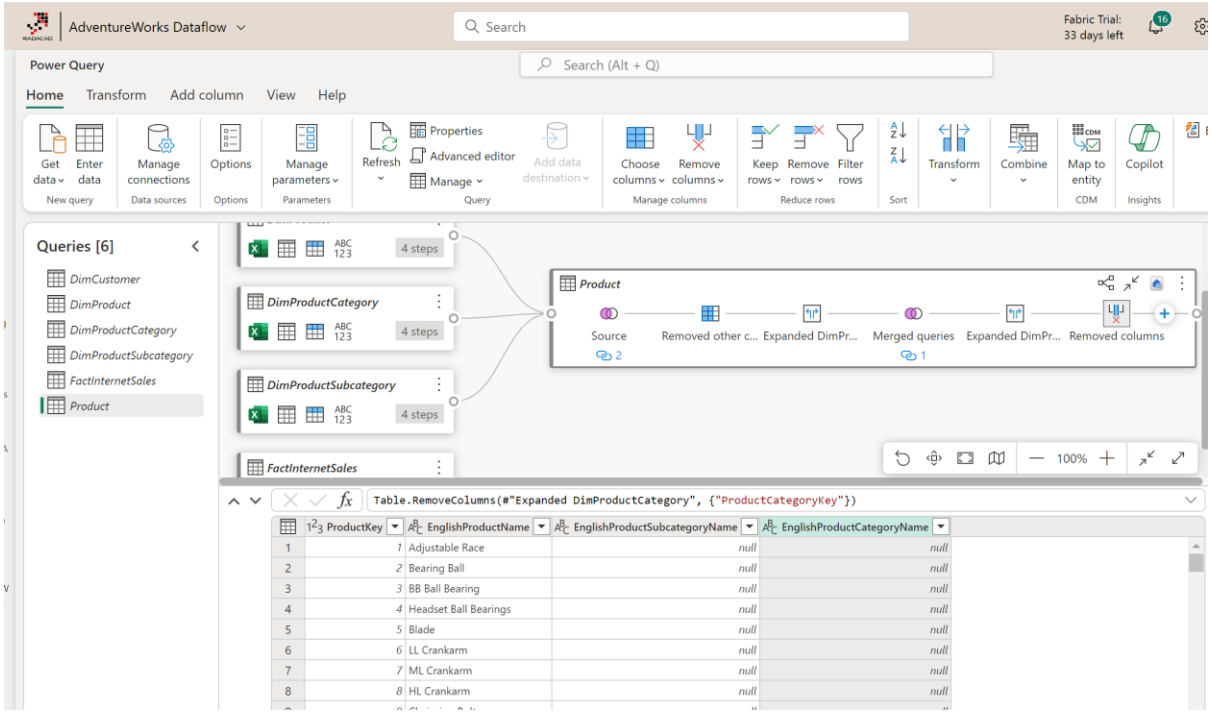
32. Once merged, you can first remove all columns, but only keep ProductKey, EnglishProductName, and this new DimProductSubcategory, and then expand it to underlying columns of ProductCategoryKey and EnglishProductSubcategoryName



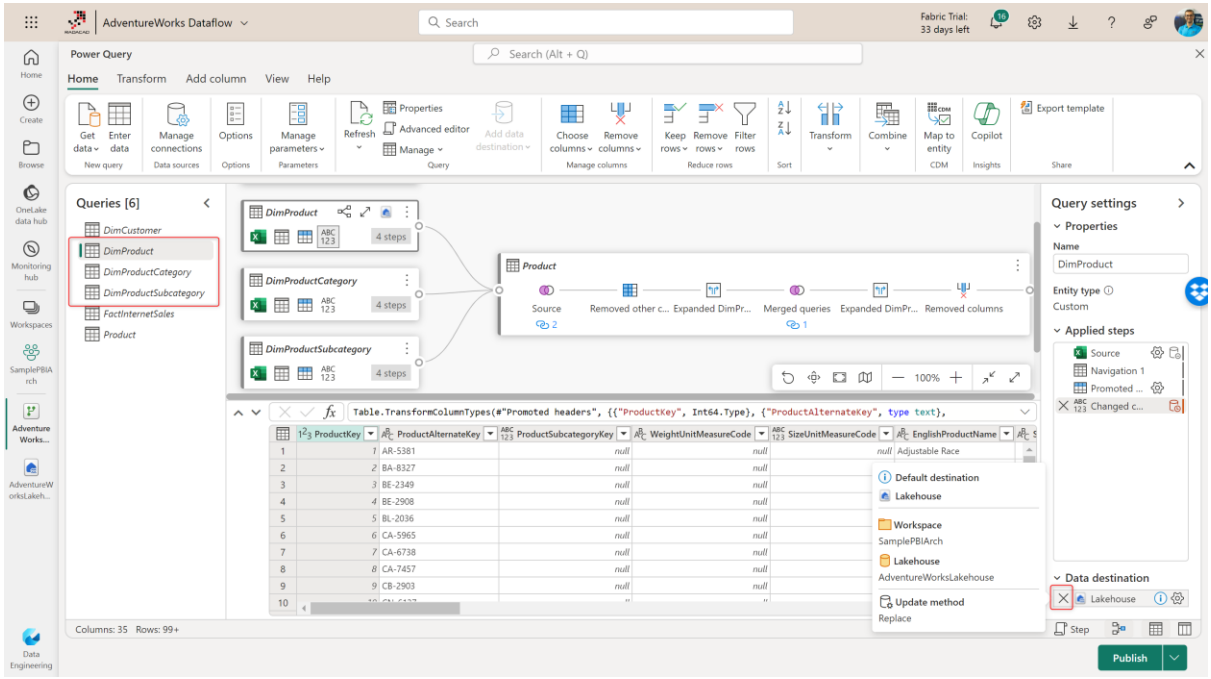
33. Merge the table one more time, but this time to DimProductCategory on the ProductCategoryKey column.



34. After the merge, Expand the result to EnglishProductCategoryName, and remove the ProductCategoryKey column. Rename the table to: Product.

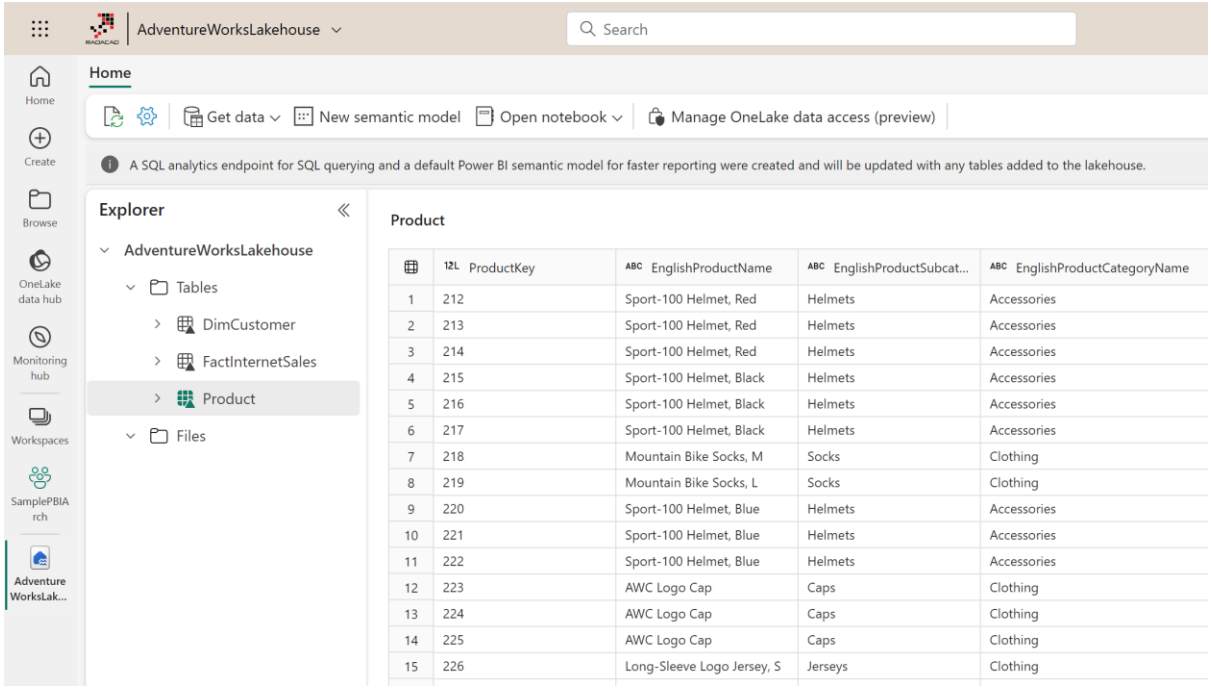


35. Disable the load of DimProduct, DimProductCategory and DimProductSubcategory, but removing their destination.

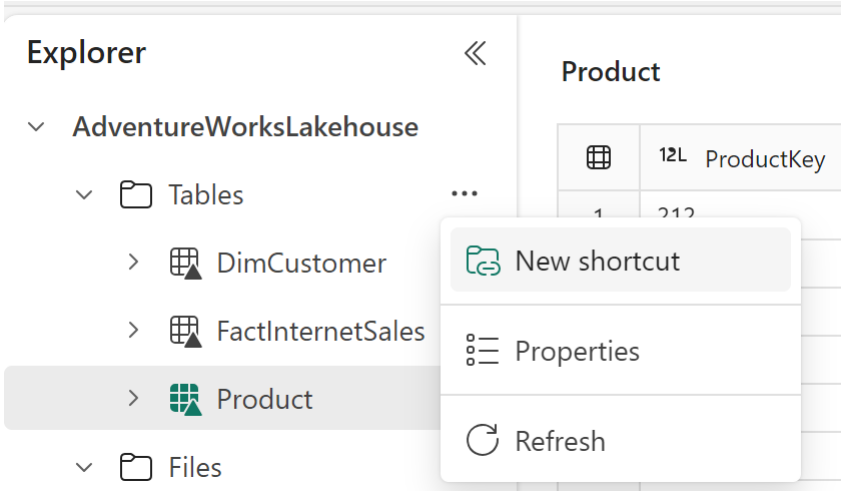


36. Publish the Dataflow, and after completing the refresh of it, check the tables in the new AdventureWorkLakehoues.
37. The tables loaded in the AdventureWorksLakehouse will be like the below screenshot;

Training@radacad.com

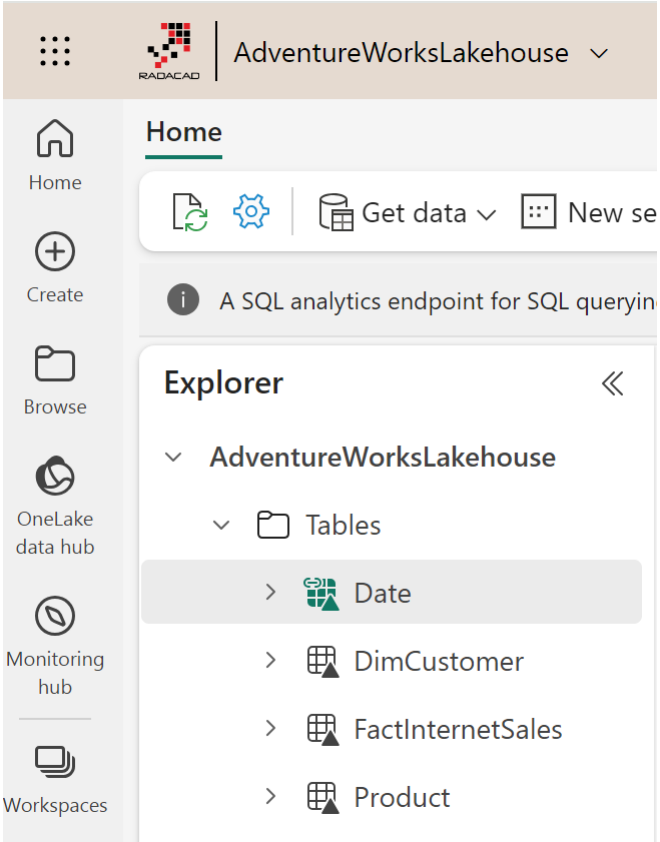


38. Create a new Shortcut in this Lakehouse.

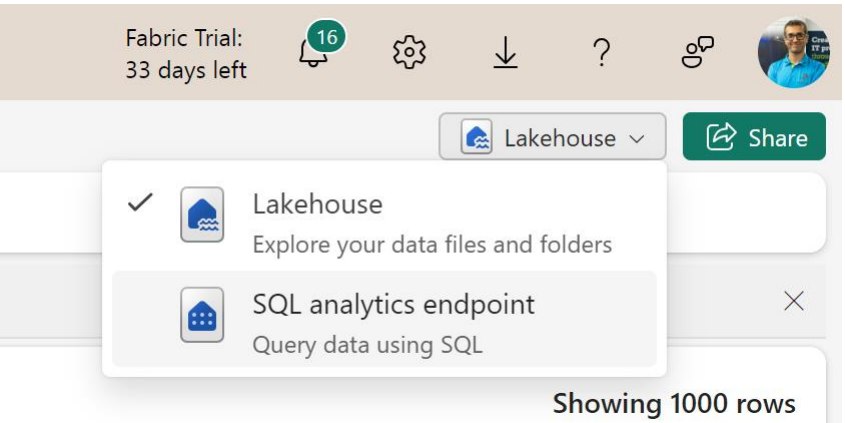


39. Target the shortcut to the GeneralLakehouse and the Date table (same as the one you did previously in the PubsLakehouse). The new Date table should now appear in the AdventureWorksLakehouse.

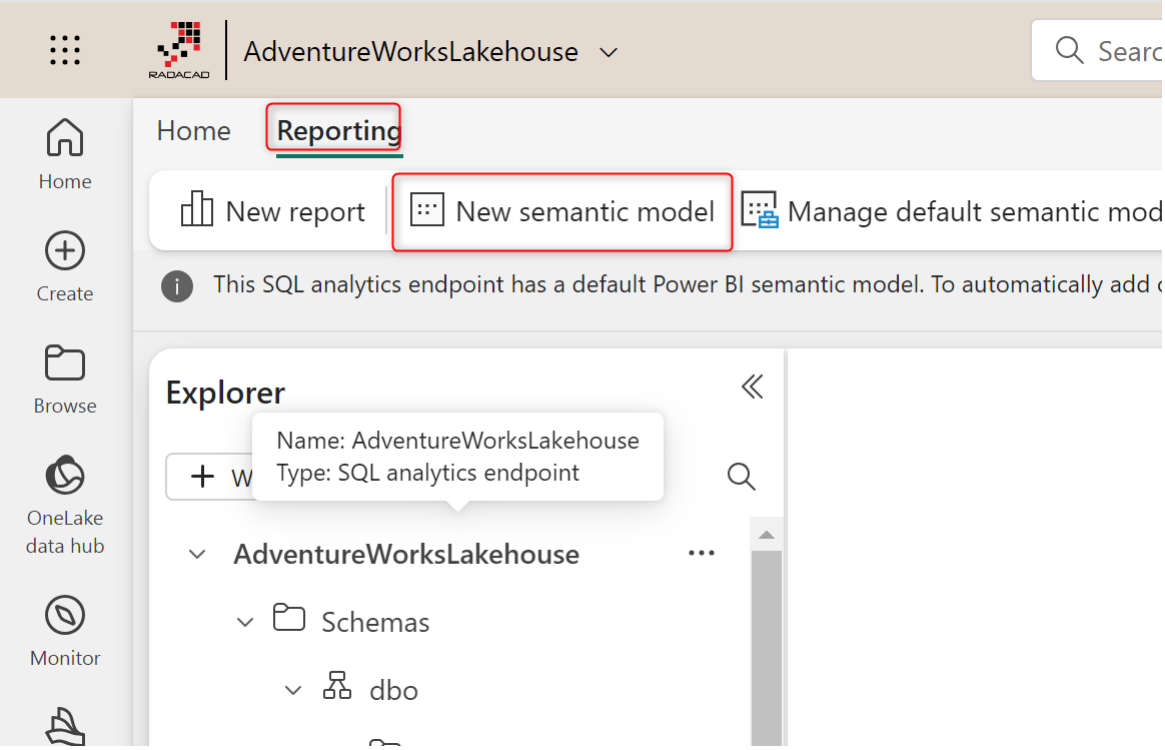
Training@radacad.com



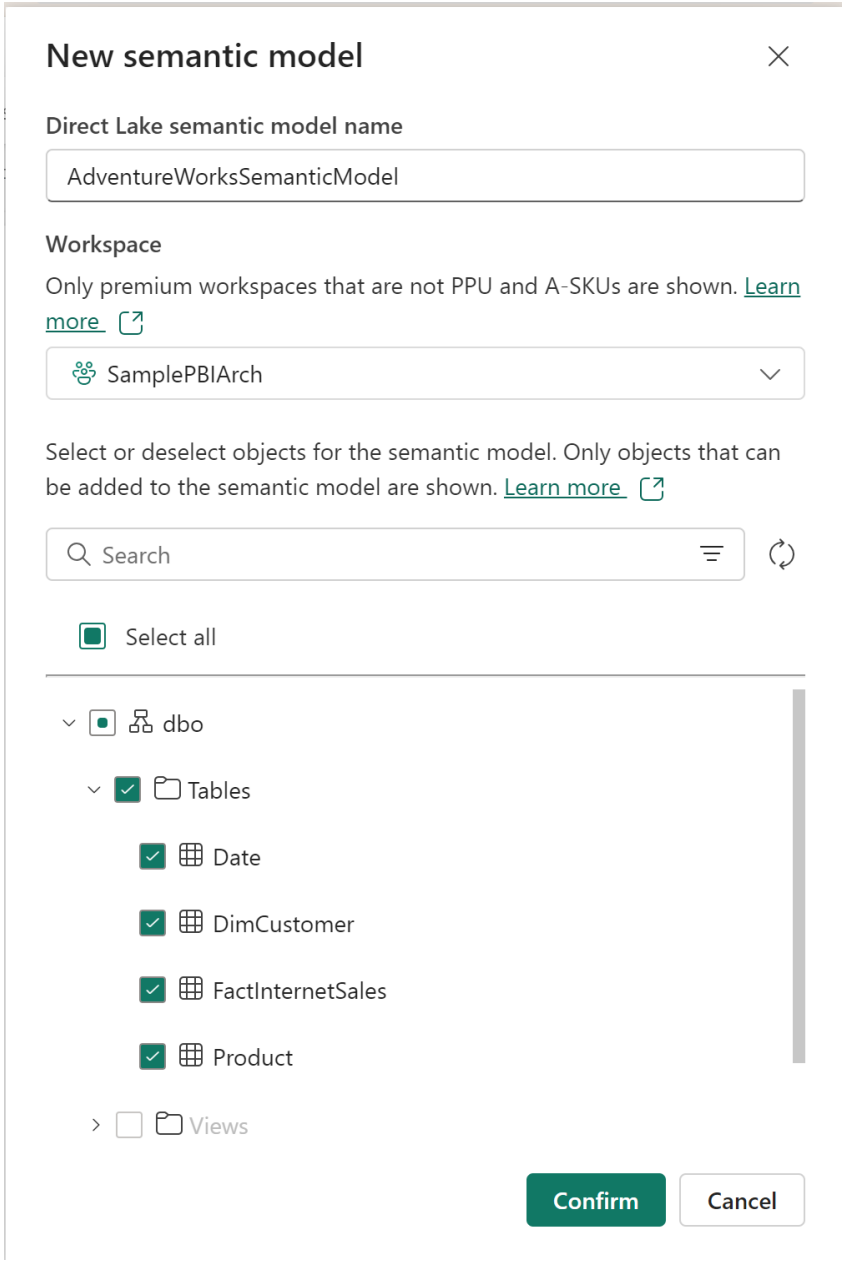
40. Switch from the Lakehouse Explorer to the SQL Analytics Endpoint



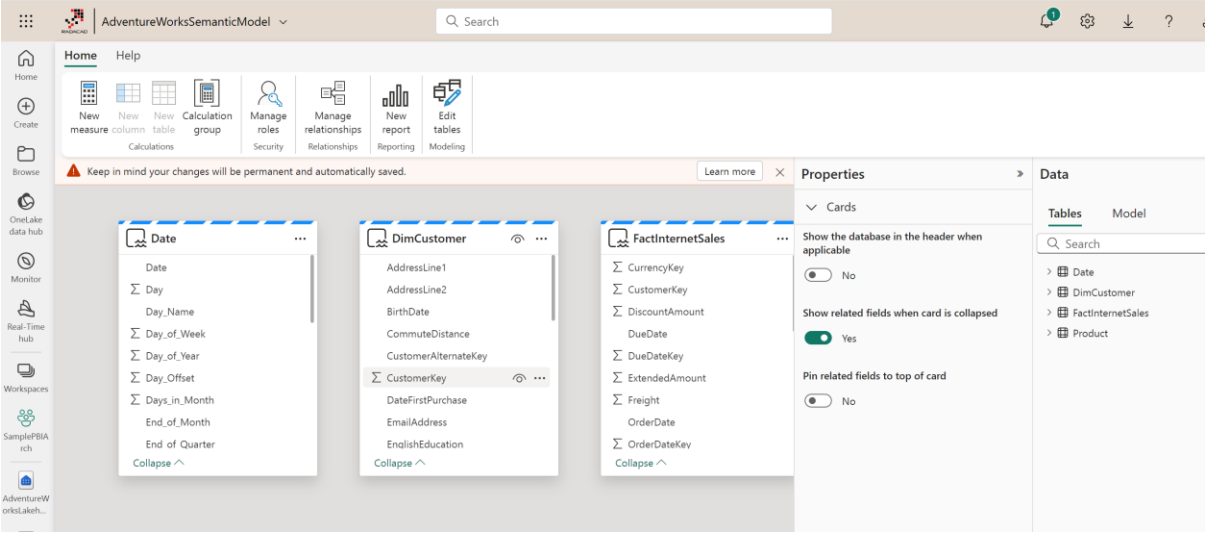
41. Go to the Reporting tab, and click on New Semantic Model



42. Name this new Sematic Model as AdventureWorksSemanticModel, and select all tables

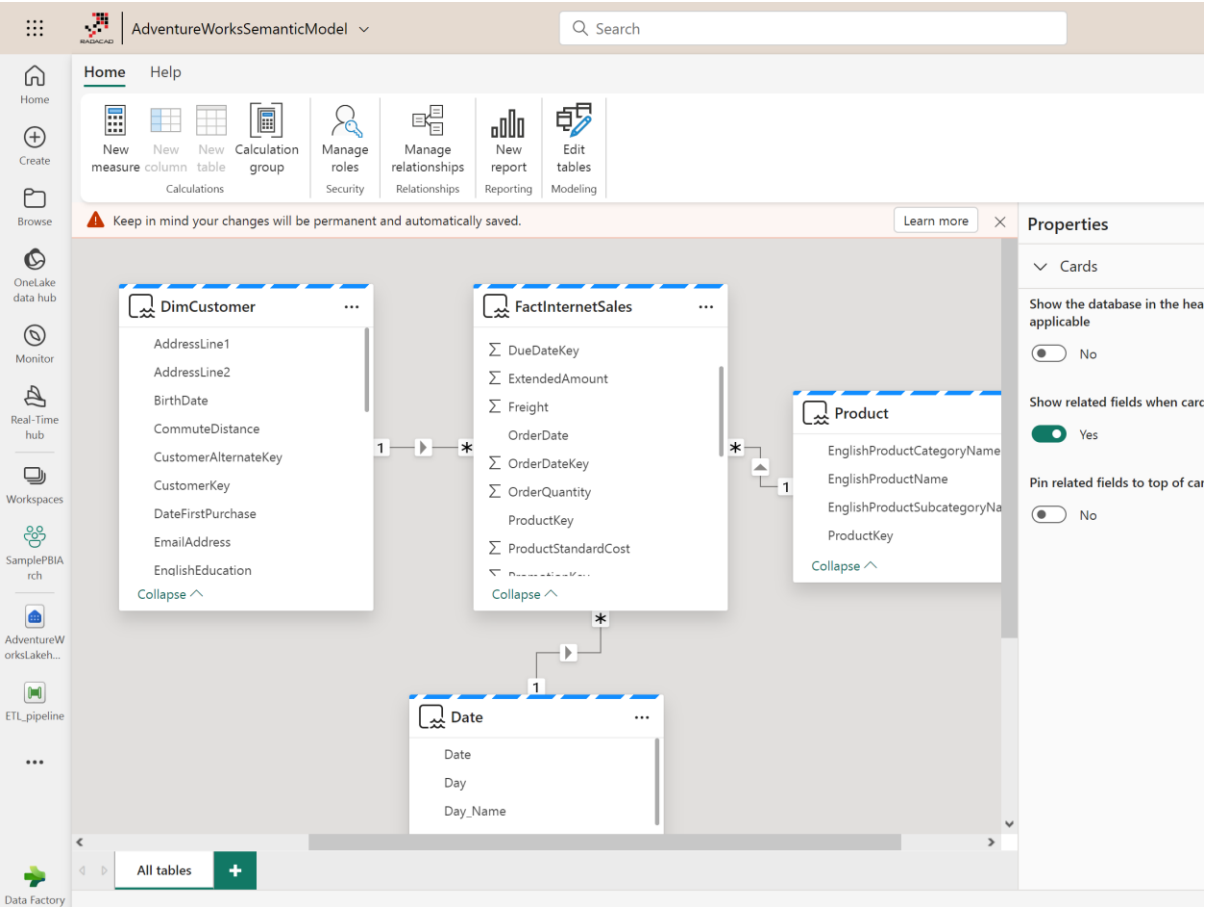


43. Once the new sematic model generated, you can see the model editor as below;



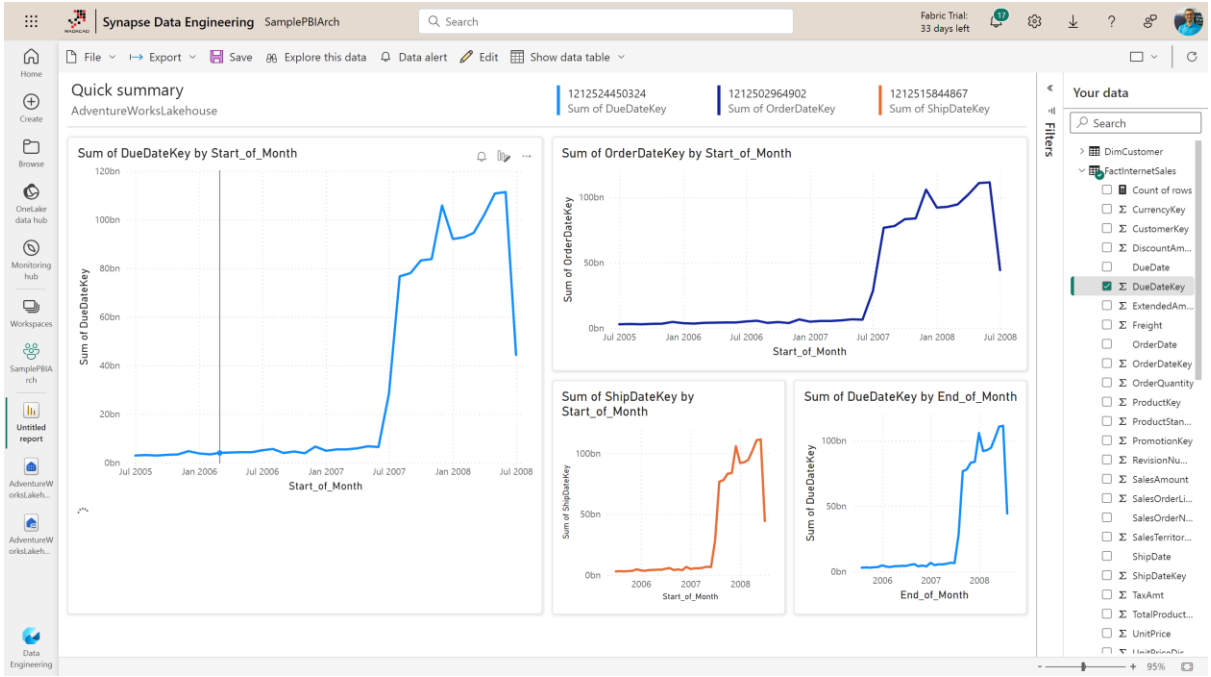
44. Create a relationship diagram below;

DimCustomer to FactInternetSales (CustomerKey)
Product to FactInternetSales (ProductKey)
Date to FactInternetSales (Date > Order Date)



45. Then create a new report from this model, here is an example report;

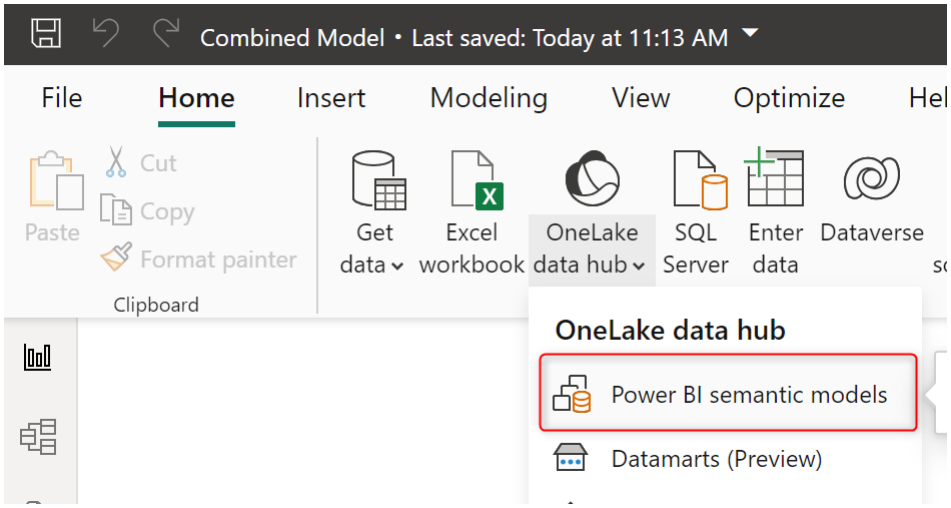
Training@radacad.com



46. Save the report in the same workspace.

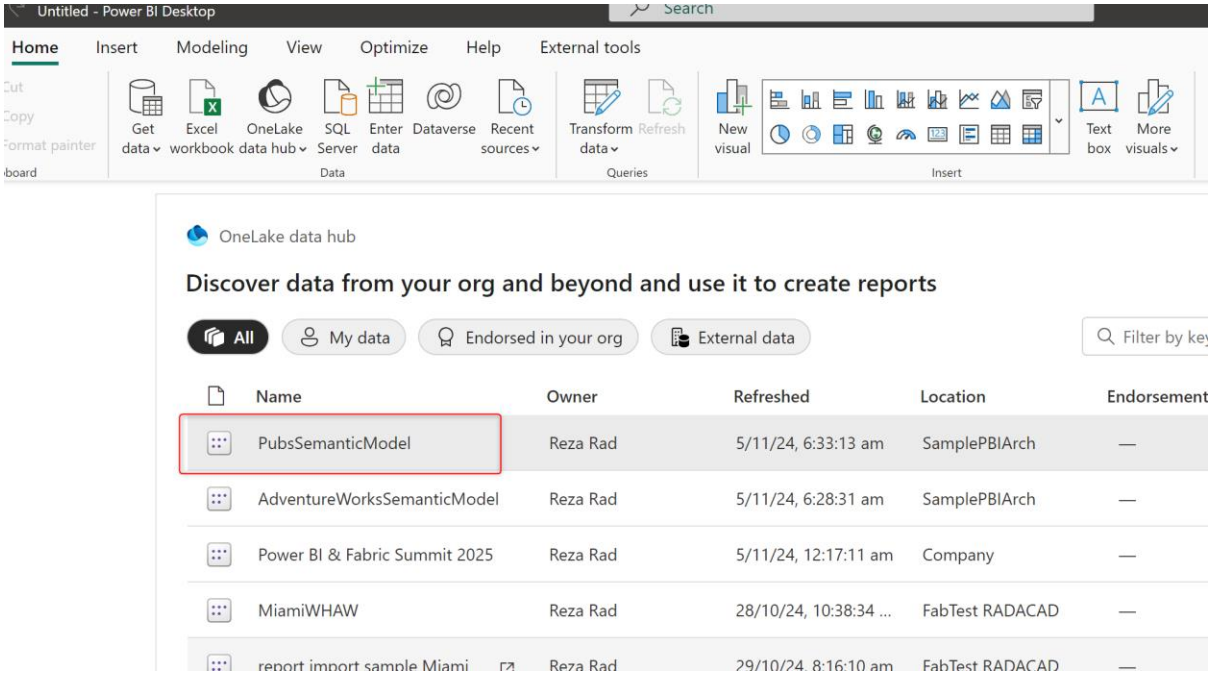
Stage 4: Composite Model

47. Open Power BI Desktop and get data from Power BI Semantic Model.



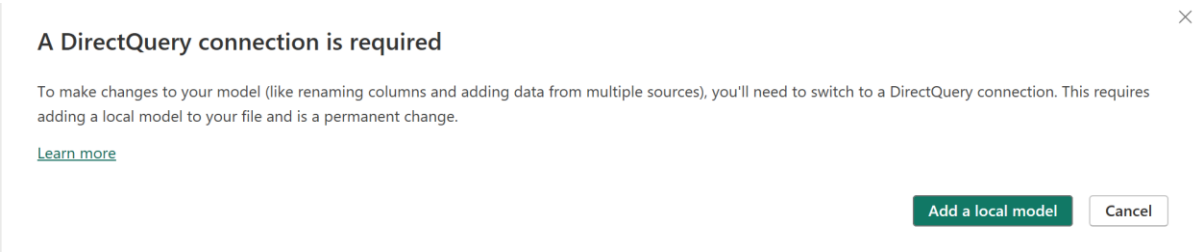
48. Get data from the PubsSemanticModel.

Training@radacad.com

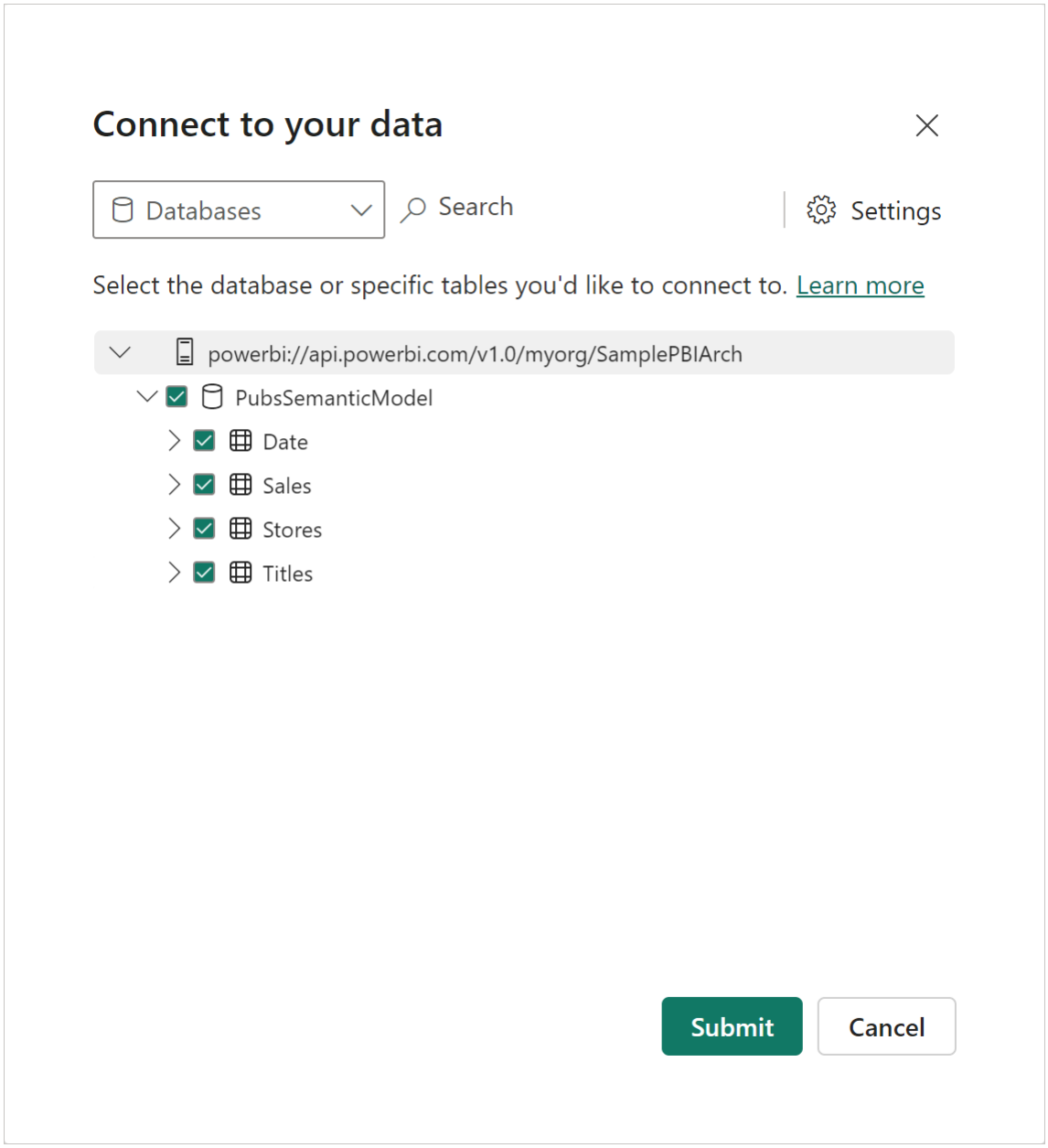


49. Once connected, Note that at the bottom of the report it shows Live connection to the PubsSemanticModel.

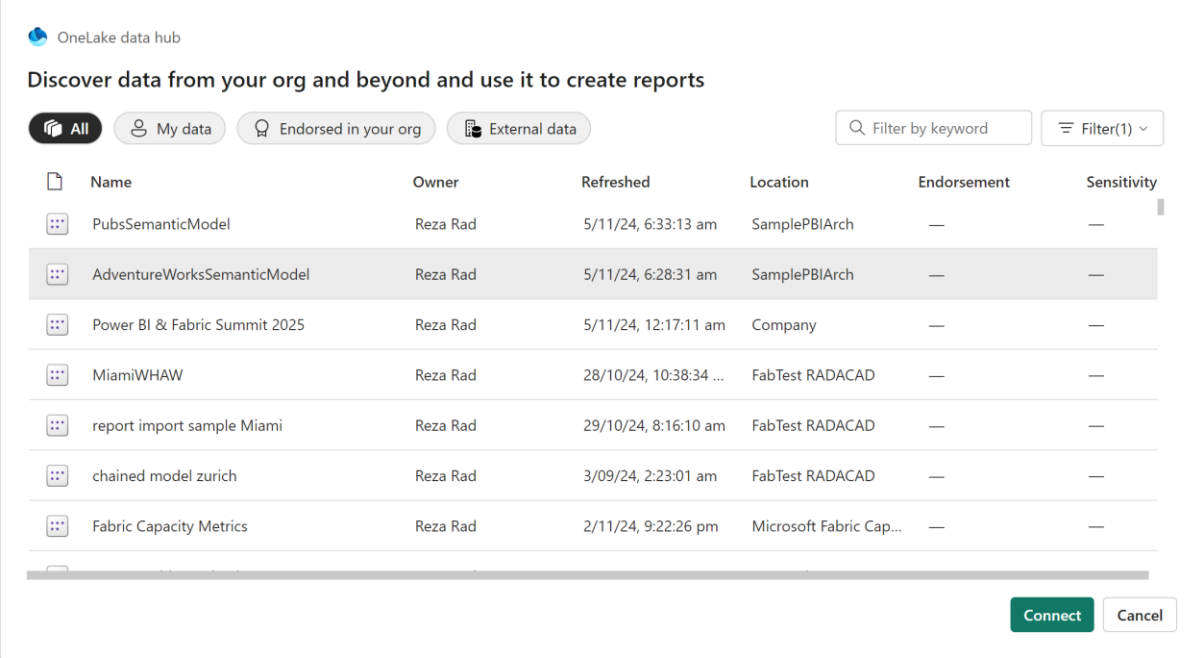
50. click on get data from Semantic model again, and this time you will be asked to create a local model (which will be the Composite model);



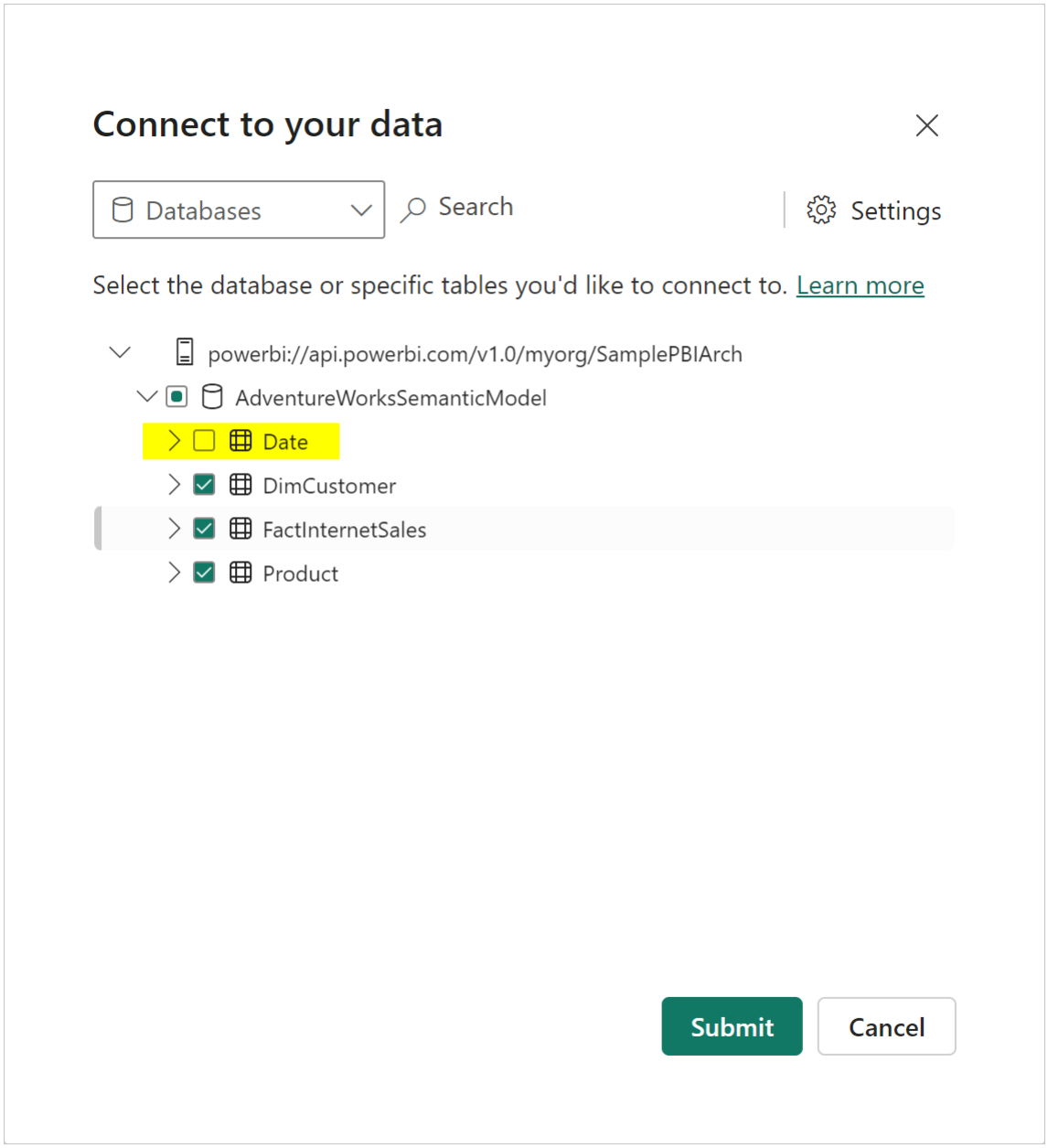
51. After selecting that, you can choose all the tables from the Pubs model



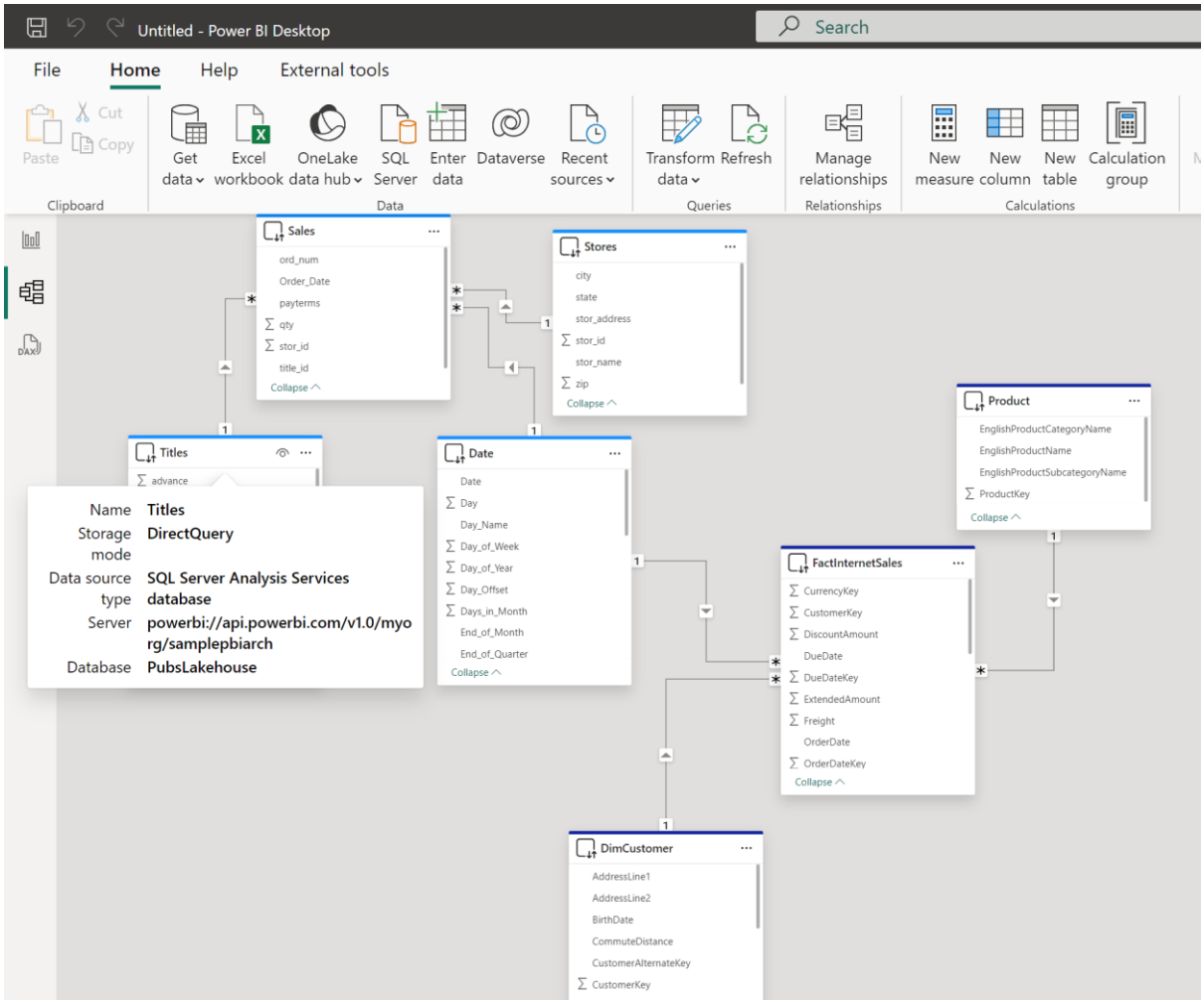
52. Note that the storage mode in the Power BI Desktop now became DirectQuery (Which means DirectQuery to the Power BI semantic model). Select AdventureWorksSemanticModel as the next model to connect.



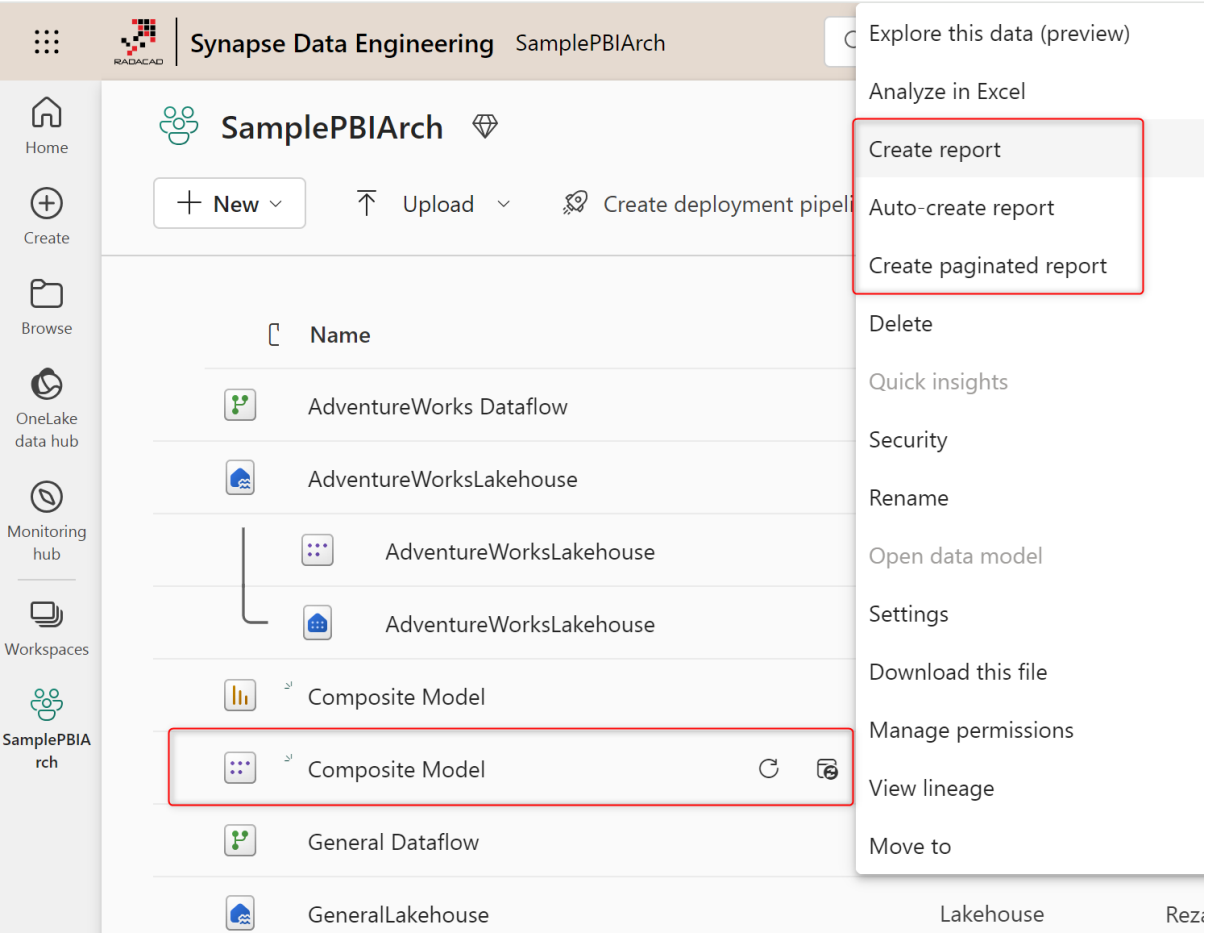
53. Select all the tables except Date, because you already have that one in the model from PubsSemanticModel.



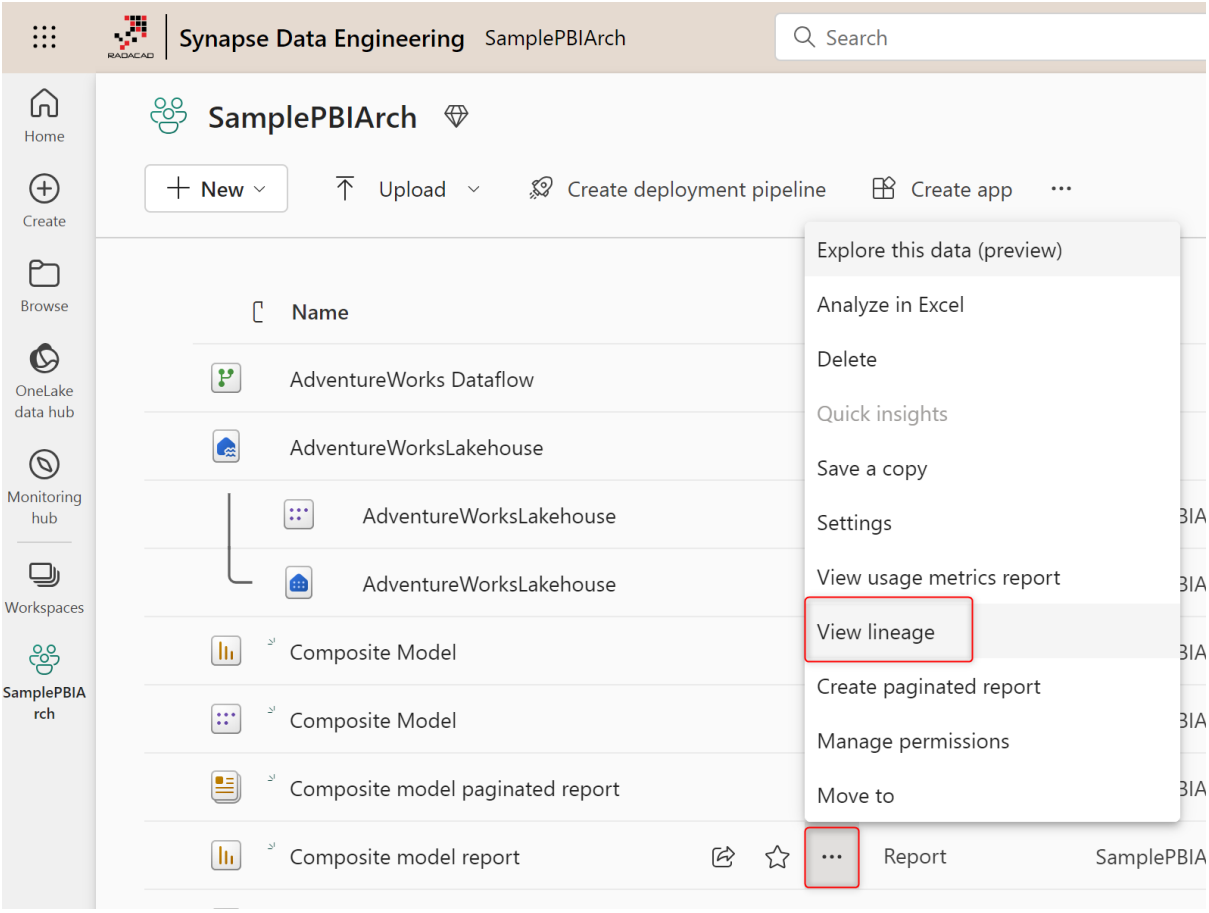
54. Once the connection is created, go to the model tab, and create a relationship between the Date table (Date field) and FactInternetSales table (Order Date field). Note also the color code of table headers based on where the data is coming from. You can also hover the mouse on that colored table header bar and see the information.



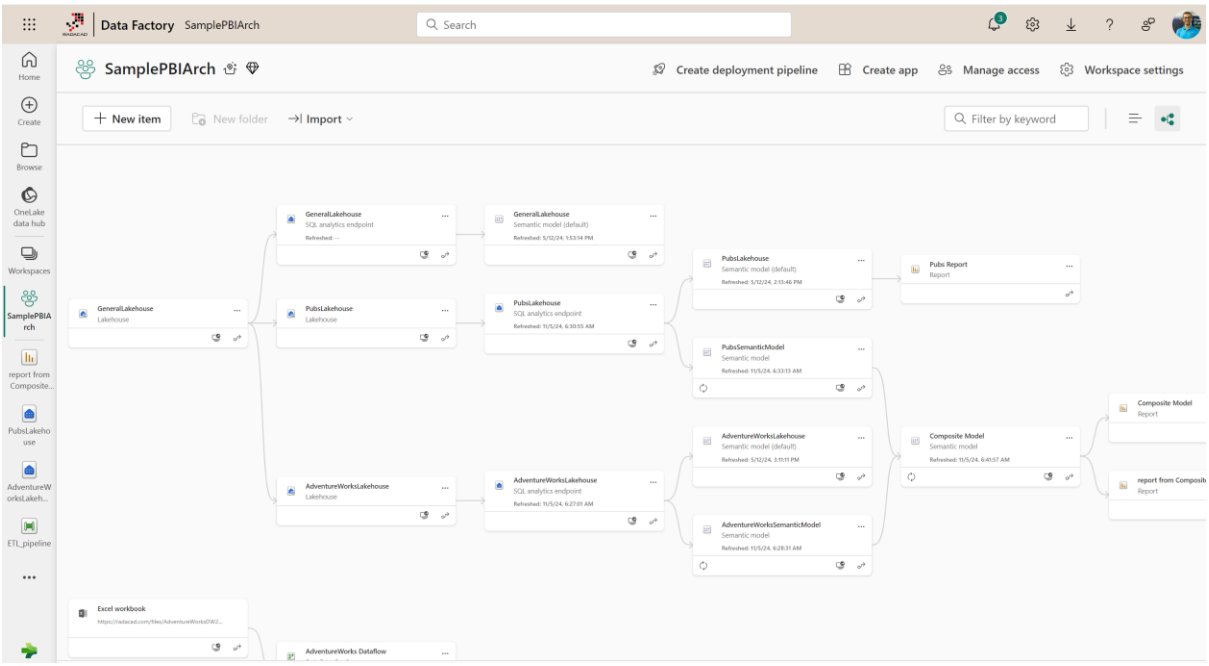
55. Once done, Save the Power BI file as Composite Model, and Publish it to the same workspace.
56. In the Power BI Service, click on more options on this new Semantic Model and create a report from it.



57. You can also create a paginated report if you want, having two types of report will be good to see they use the same shared semantic model.
58. On the newly generated report, click on the more options, and choose View Lineage



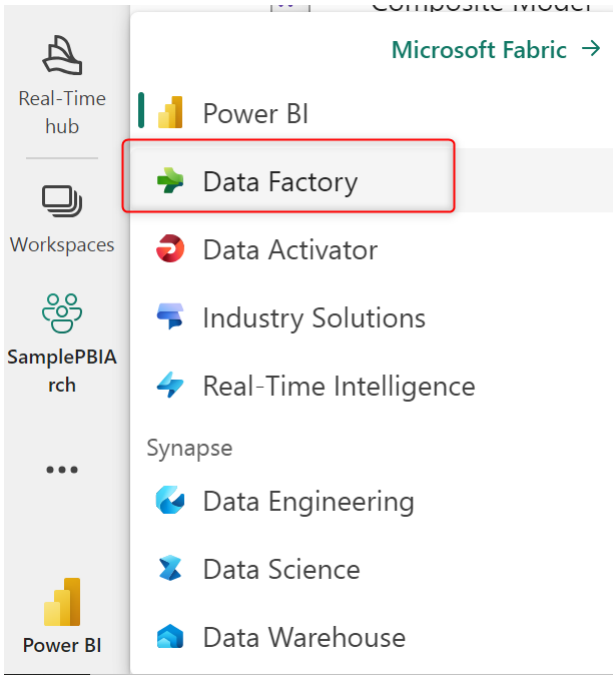
59. The full Lineage of items all the way from the General Dataflow to each Lakehouse and then to the composite model will be shown here.



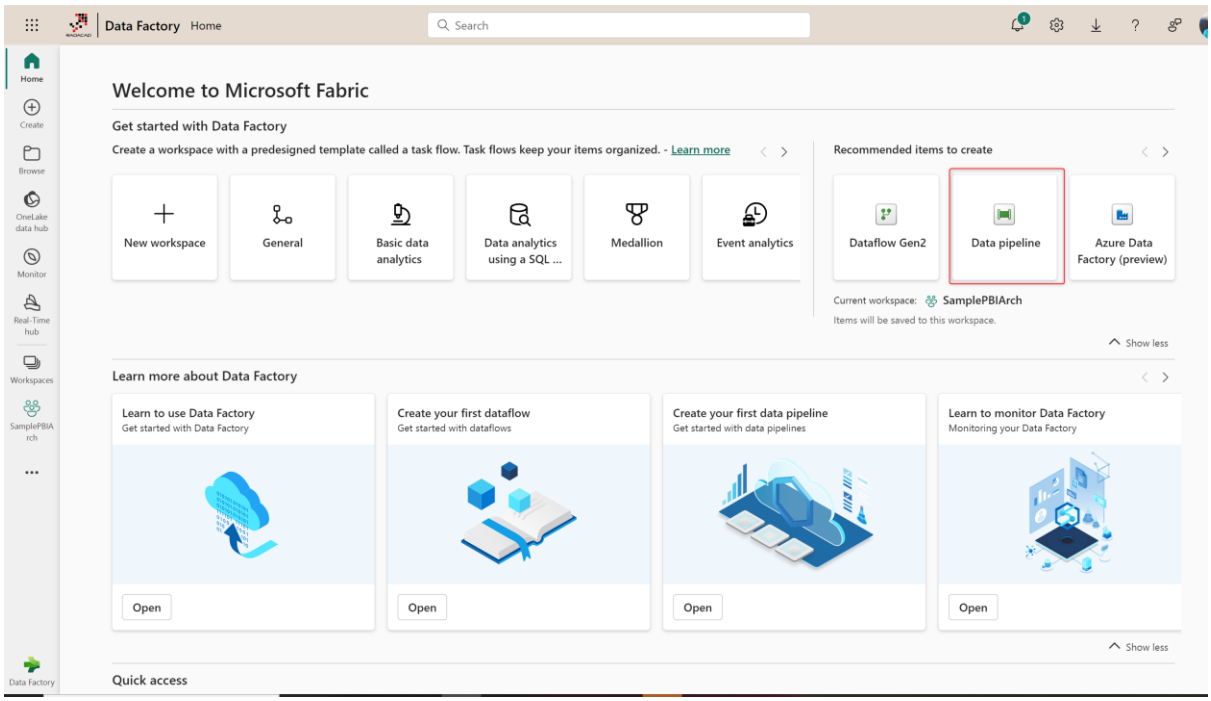
Training@radacad.com

Create a pipeline to streamline the refresh of data

60. Navigate to Data Factory workload

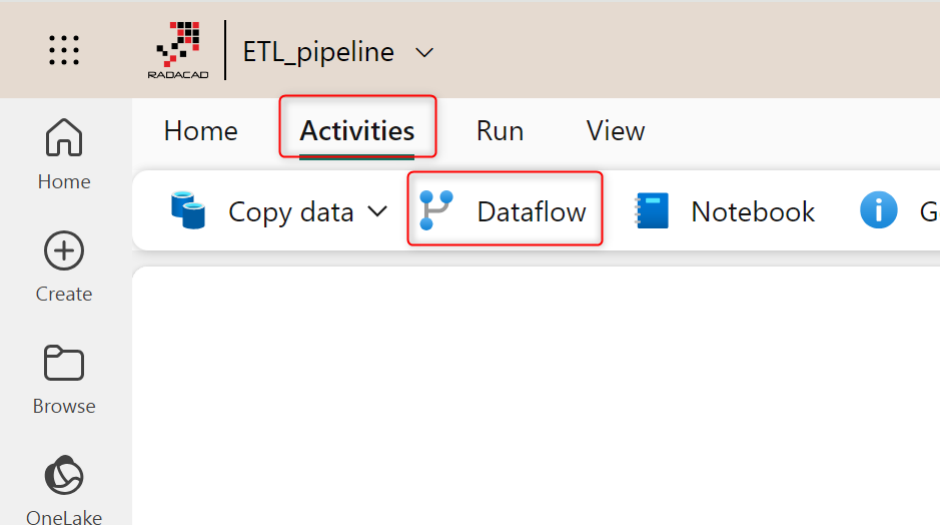


61. Create a Data Pipeline, and name it ETL_Pipeline

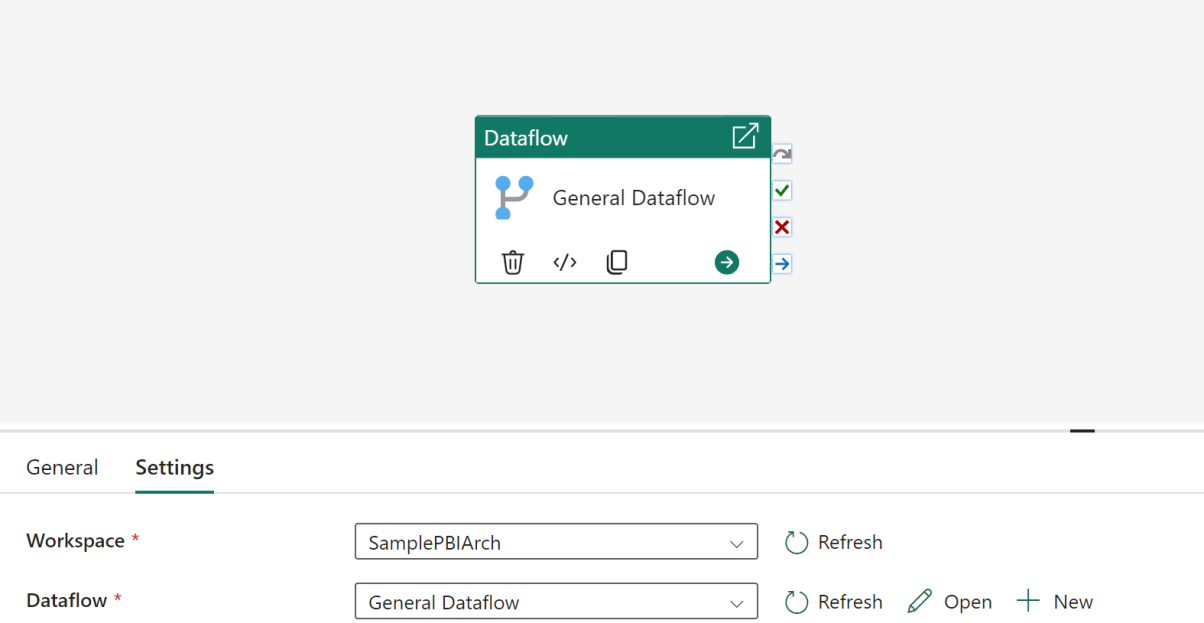


62. In the pipeline editor navigate to Activities tab, and add Dataflow

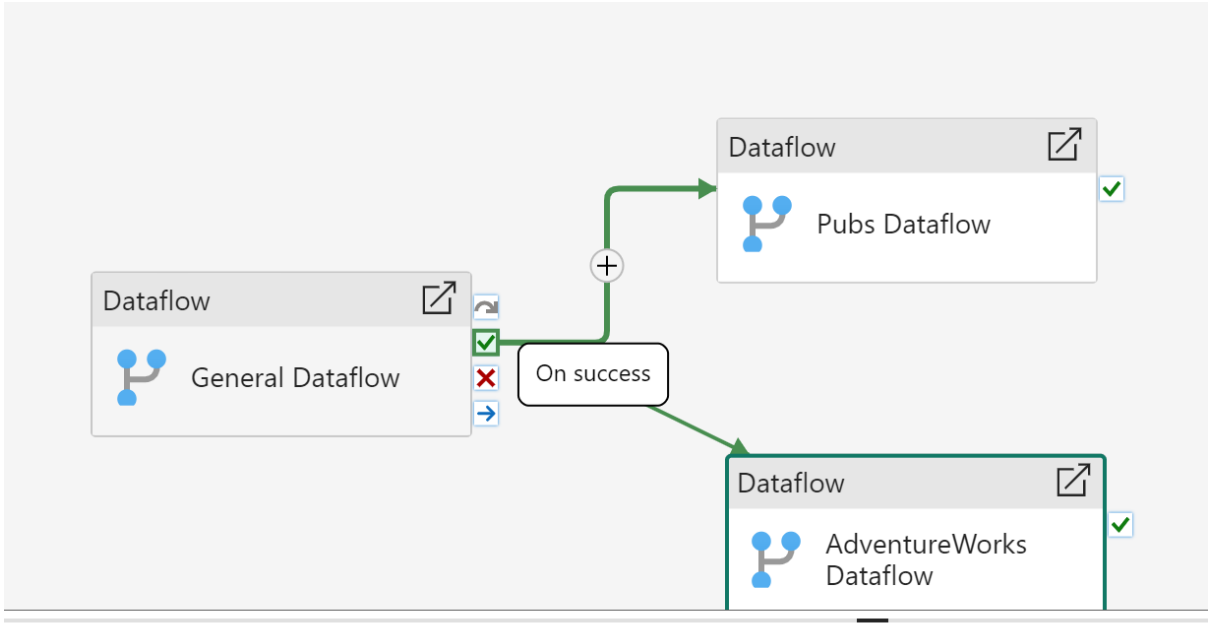
Training@radacad.com



63. In the settings of Dataflow activity select General Dataflow. Also name this activity as General Dataflow.

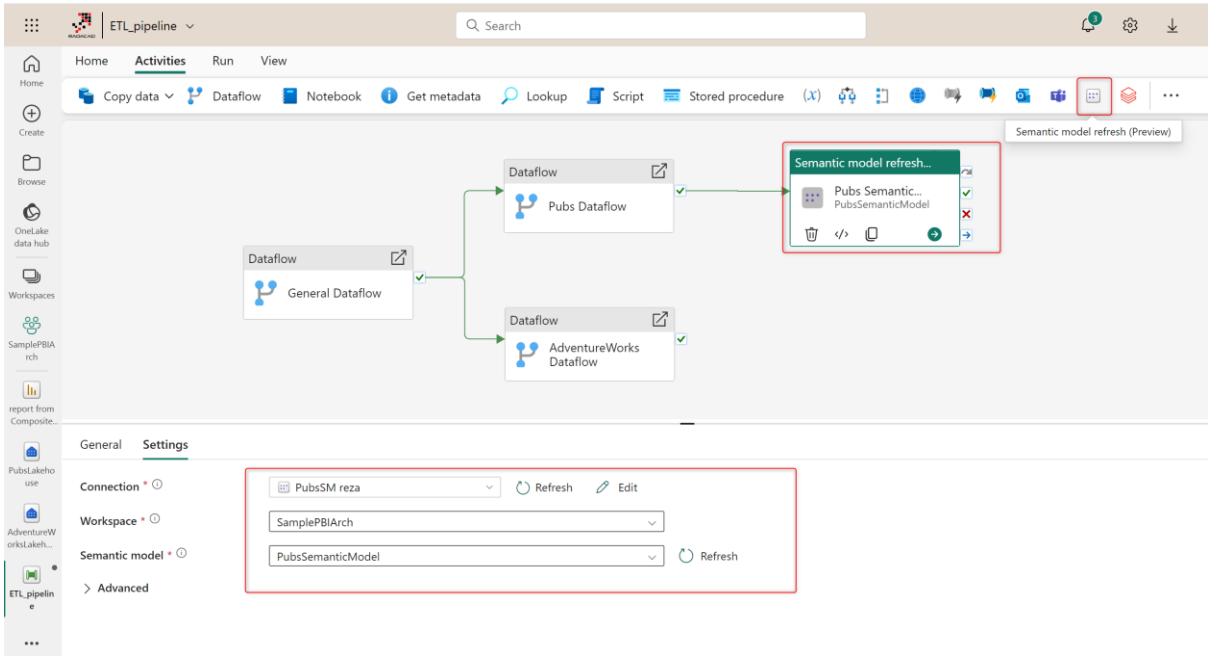


64. Add two other dataflow Activities and connect the Success output from General Dataflow to them, these two new dataflows will run the Dataflows for Pubs and AdventureWorks.

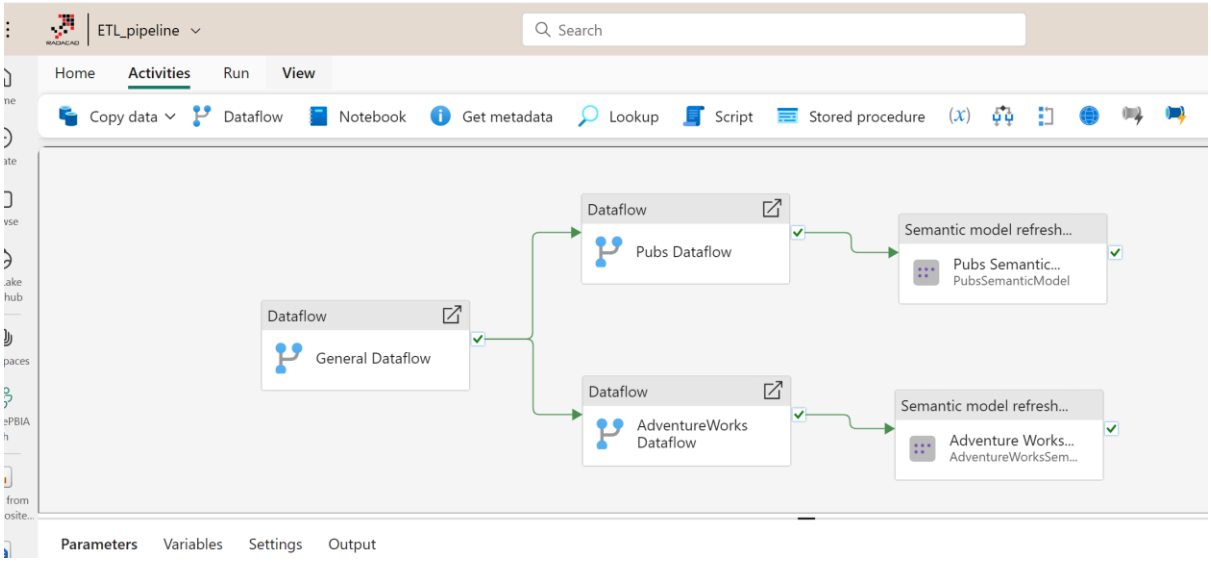


Output

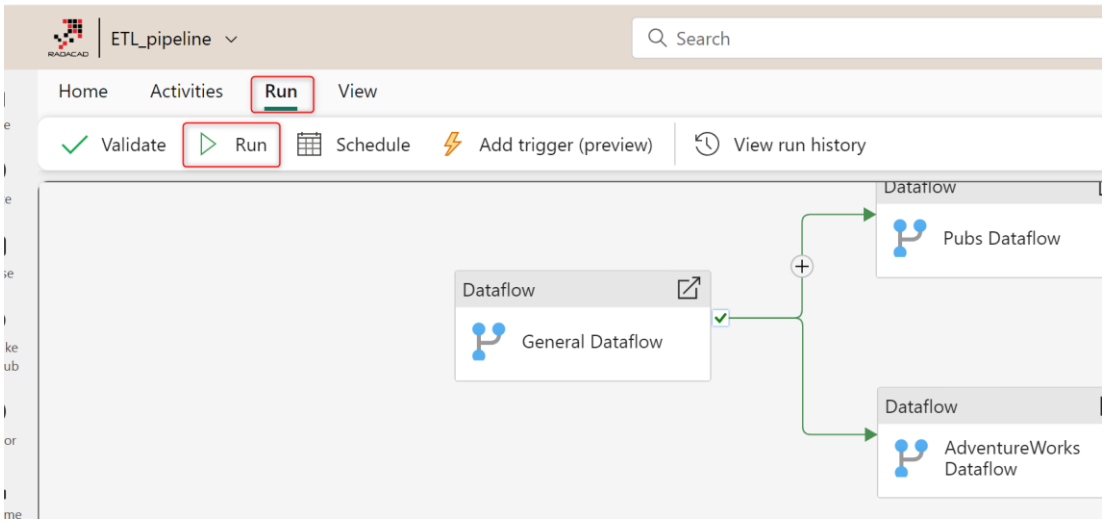
65. You can add the semantic model refresh also after each dataflow



66. Although in this scenario the semantic models are Direct Lake and they don't need refresh, but the process is shown in case you want the refresh to be done through the pipeline.



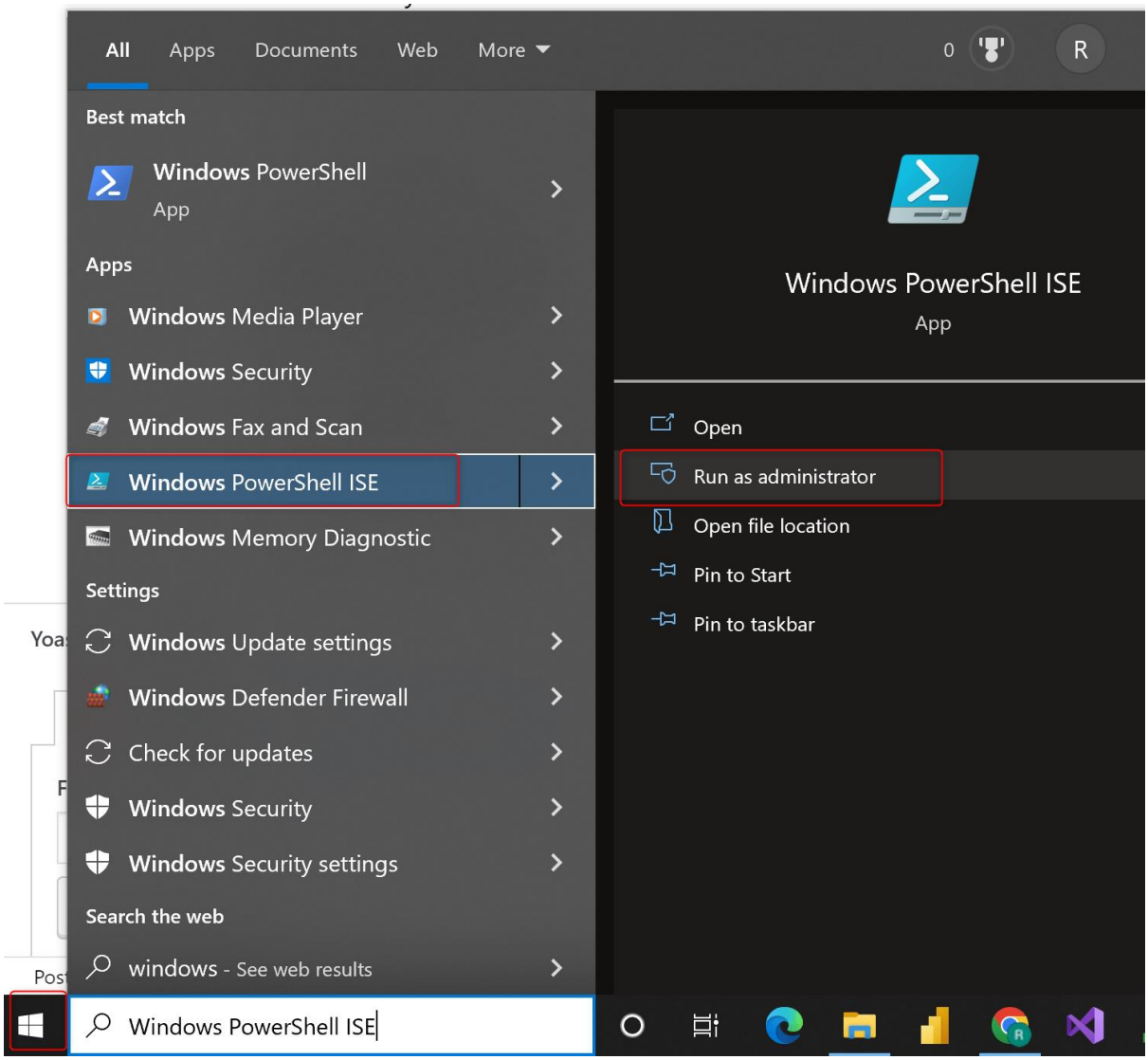
67. Save and run the pipeline



Training@radacad.com

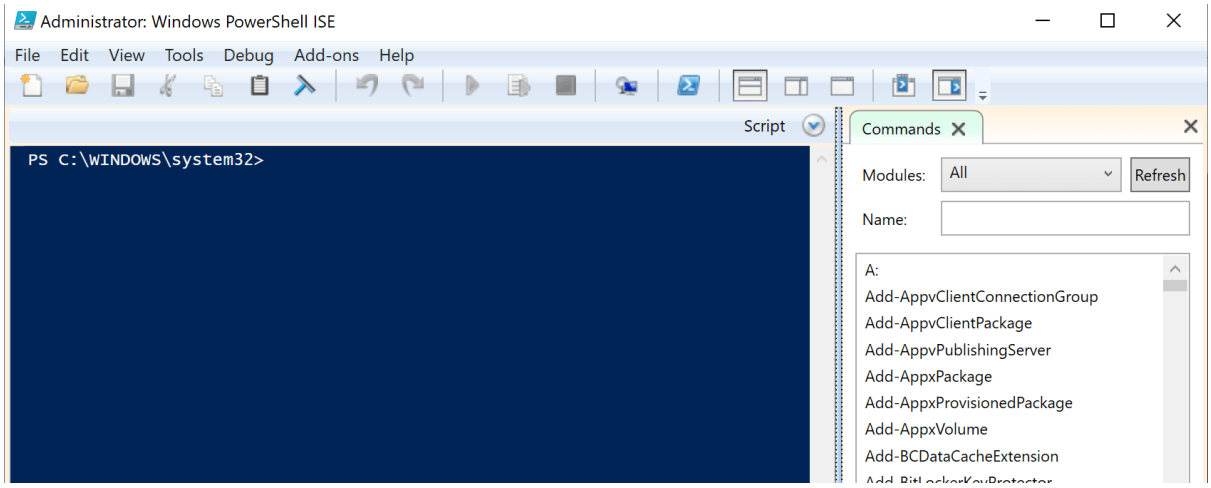
Lab Exercise 2: Powershell Cmdlets for Power BI

1. To start, open Windows PowerShell ISE (this is a more user-friendly version of Windows PowerShell, If you don't have this application, use the normal Windows PowerShell). It is better if you start this application as an administrator.



Opening Windows PowerShell ISE as administrator

2. This will open the application below, in which you can either type the commands (also called Cmdlets) or use the list on the right-hand side and select from the cmdlets (this list is only available in ISE, the basic Windows PowerShell doesn't come with it).

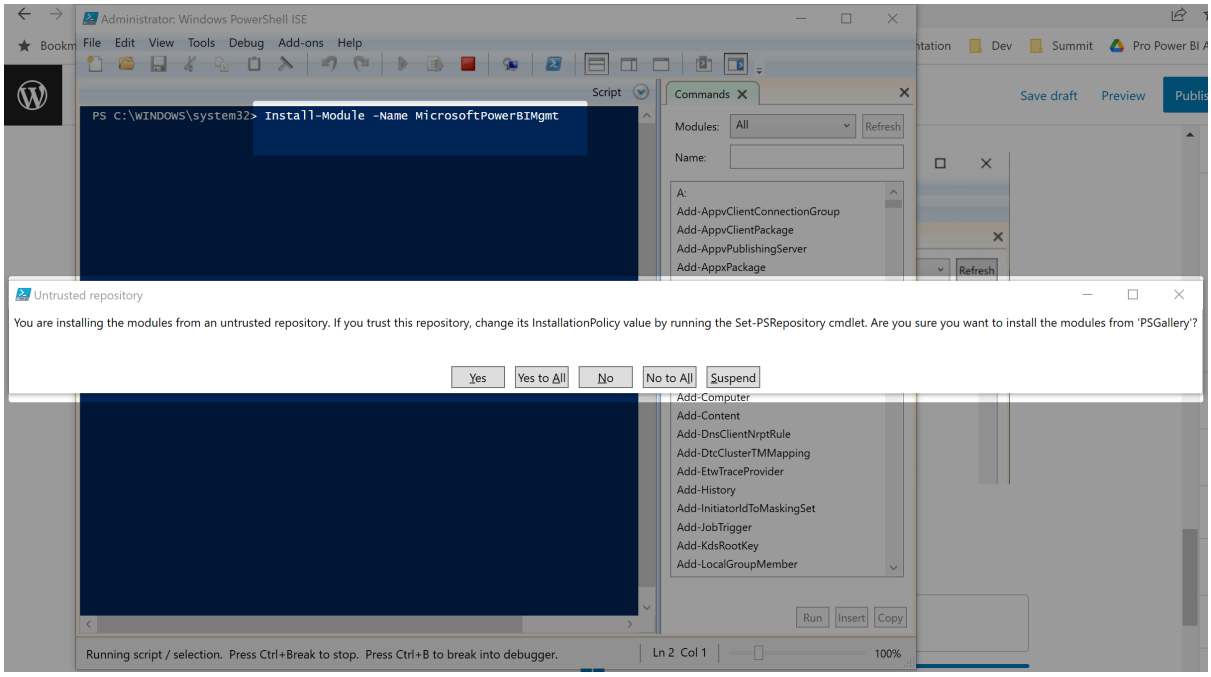


Windows PowerShell ISE

3. To get the modules to install, you can use the Install-Module Cmdlet as below;

Install-Module -Name MicrosoftPowerBIMgmt

Then confirm the installation of the module

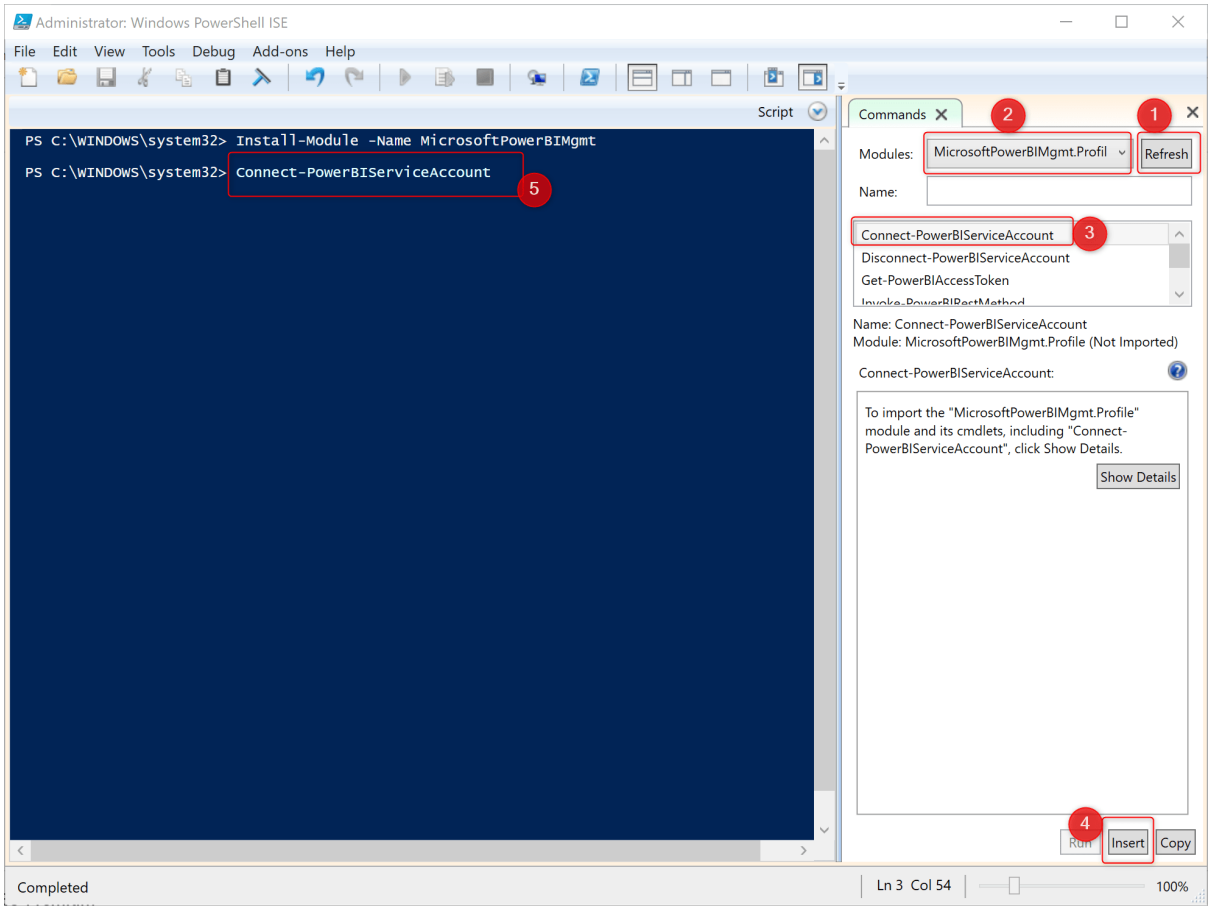


Installing Power BI modules for PowerShell

4. This will install the module. If you already have the module installed and want to update it to the most recent version, use the Update-Module Cmdlets with the same parameters.
5. To use the Power BI Cmdlets, first, you need to log in to Power BI using Connect-PowerBIServiceAccount Cmdlet as below;

Connect-PowerBIServiceAccount

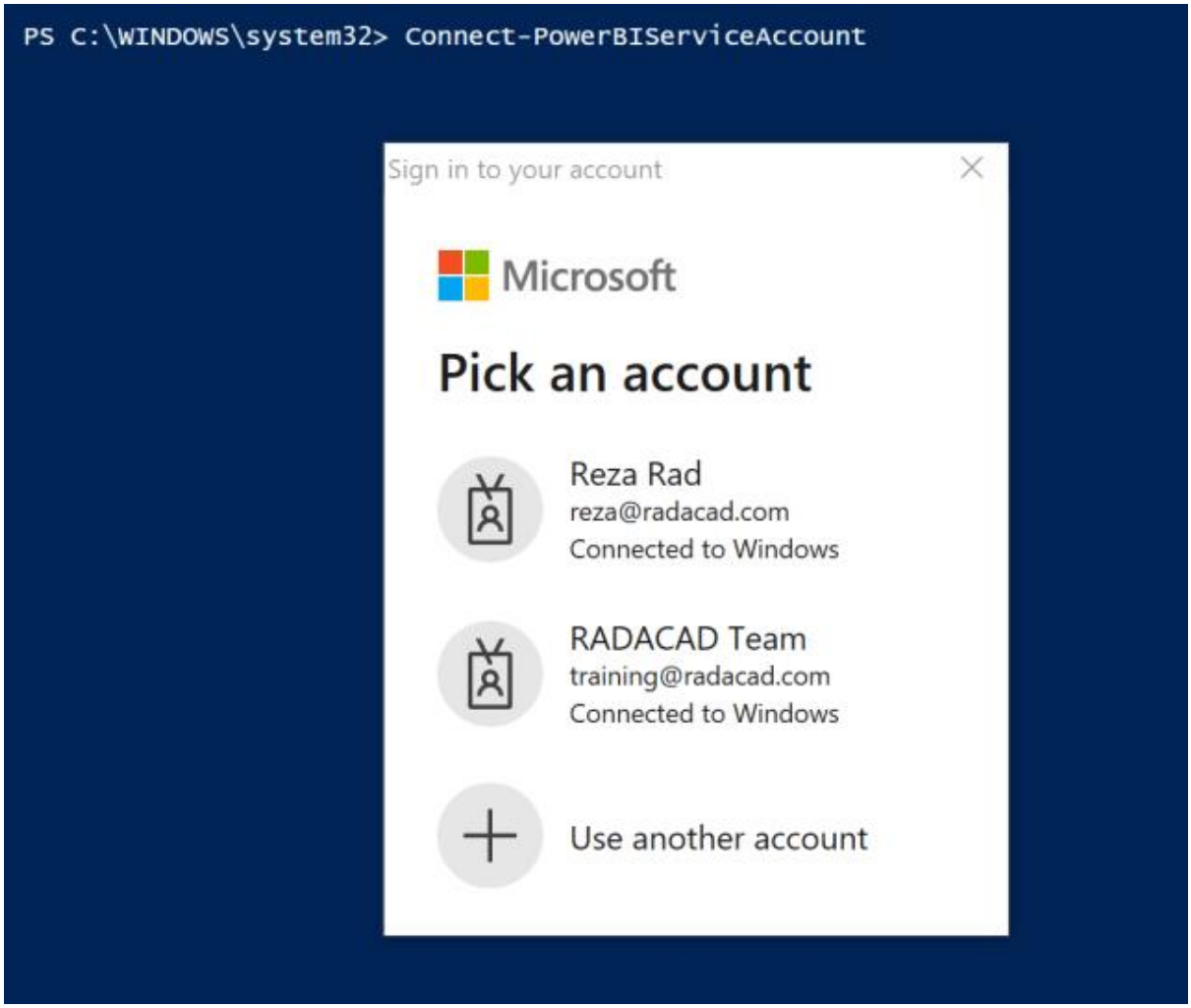
Or, if you prefer to use the list of modules in the PowerShell ISE, Refresh the modules list, then select Power BI Profile module, and select the Cmdlet under that. You can either Run the command from here or insert it, which will insert in the command line.



Using the list of Cmdlets in PowerShell ISE

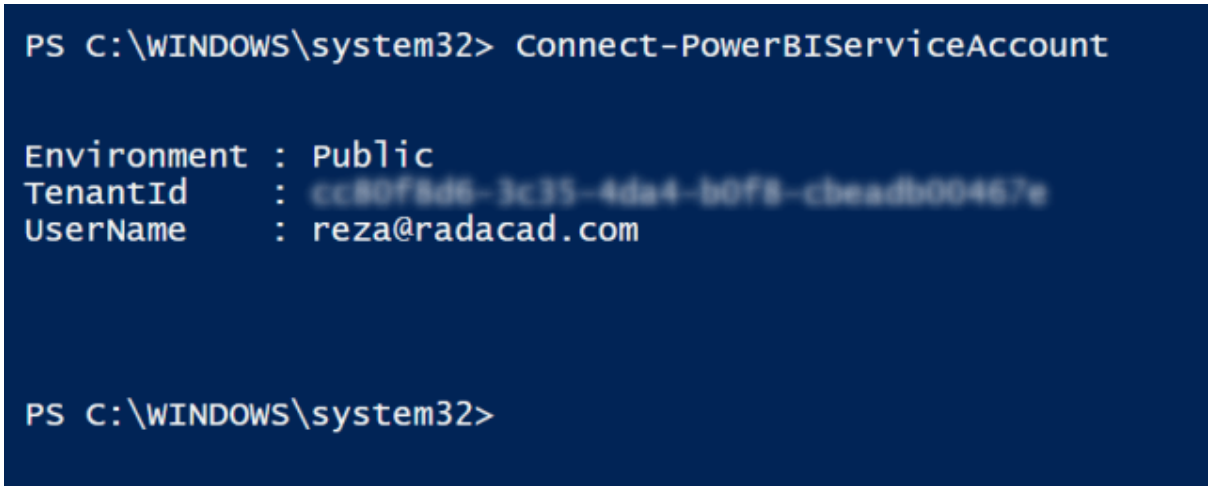
If you choose a Cmdlet that requires parameters, the UI will also have the places to enter those details.

6. Use the sign-in box to use your account for the Power BI. If your account is a Power BI administrator, it might give you tenant results in your command execution. Otherwise, you can still use your non-admin Power BI account to use some of the commands.



Login using the Power BI Account

7. If the login is successful, you will see your tenant Id and the username.



The output of Connect-PowerBIServiceAccount Cmdlet

8. Now you can test some of the Cmdlets and see the result. For example, the Cmdlet below gives you a list of all workspaces (these would be workspaces that you have access to, not all workspaces in the organization);

Get-PowerBIWorkspace -All

The output will be a list of workspaces, their names with IDs, and other information.

```
Id                    : 63678ecc-f862-4fb4-a463-a1e72badb8fa
Name                  : Community
Type                  : Workspace
IsReadOnly            : False
IsOrphaned            : False
IsOnDedicatedCapacity : False
CapacityId            :

Id                    : d8aeefb1-1052-476b-9d0e-dcddde69b623
Name                  : PW
Type                  : Workspace
IsReadOnly            : False
IsOrphaned            : False
IsOnDedicatedCapacity : True
CapacityId            : 6C533637-8A43-4B09-88CF-67D689A04CBB

Id                    : d3847ea9-b624-48c1-b025-37b1bff5ede1
Name                  : PW2
Type                  : Workspace
IsReadOnly            : False
IsOrphaned            : False
IsOnDedicatedCapacity : True
CapacityId            : 6C533637-8A43-4B09-88CF-67D689A04CBB

Id                    : 48e4d0ce-84e1-472c-a306-a23abe4cf5ad
Name                  : PPU WS
Type                  : Workspace
IsReadOnly            : False
IsOrphaned            : False
IsOnDedicatedCapacity : True
CapacityId            : 6C533637-8A43-4B09-88CF-67D689A04CBB

Id                    : 36f1f927-13c8-4c1c-a4aa-d0b2bdd7ecca
Name                  : WS DIS
Type                  : Workspace
IsReadOnly            : False
IsOrphaned            : False
IsOnDedicatedCapacity : True
CapacityId            : 6C533637-8A43-4B09-88CF-67D689A04CBB



PS C:\WINDOWS\system32>
```

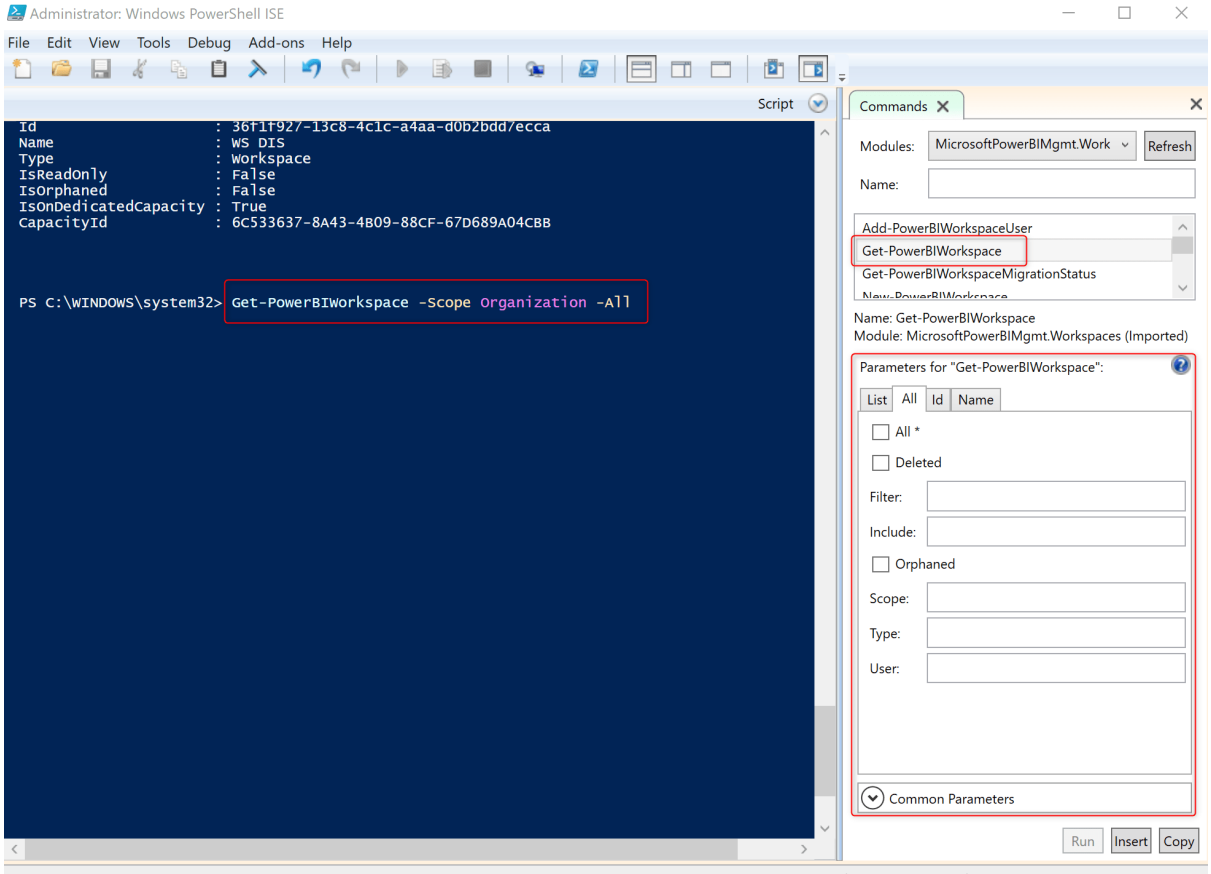
Getting the list of all workspaces for a Power BI user

Sample Scenario: Export a list of all workspaces in the organization as a CSV file

9. If your account is a Power BI Administrator account, then you can use a parameter for scope and set that to Organization, and this will give you all the workspaces in your tenant (even if you don't have access to those workspaces and other users in your organization create them);

Get-PowerBIWorkspace -Scope Organization -All

As I mentioned, you can also select this Cmdlet in the list on the right-hand side and see all other parameters available.



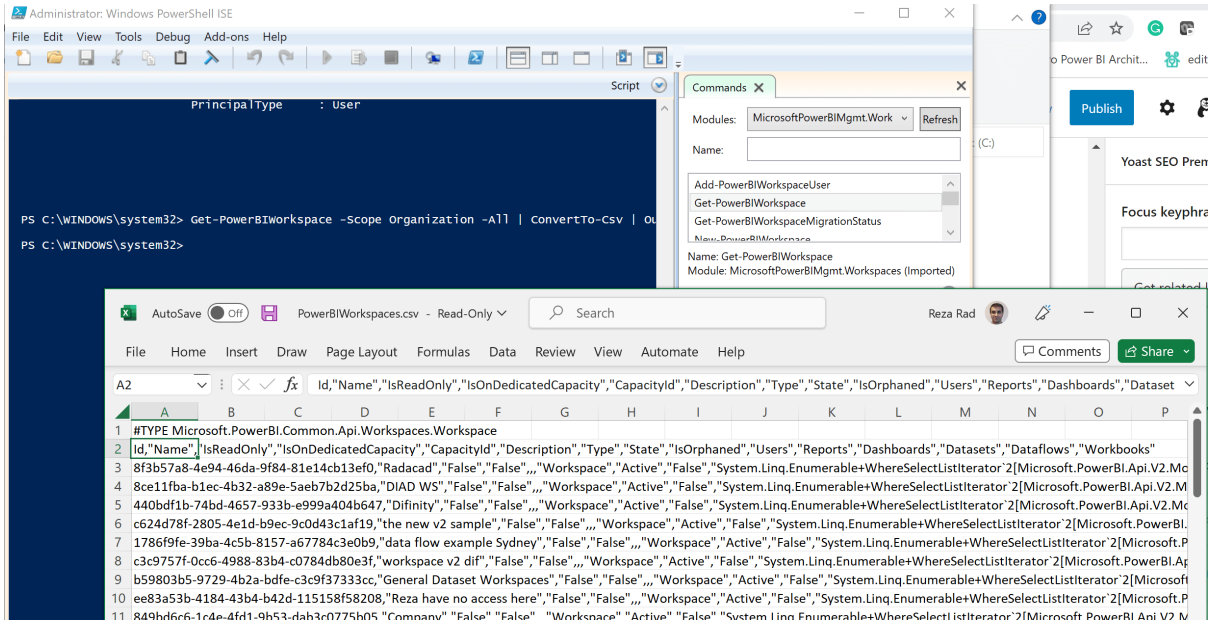
Getting all Power BI workspaces in the organization

The command above might take some time to run if you have many workspaces in your organization.

10. Now comes the power of PowerShell itself. The output above is great, but what if you could export that as a file (which you can use for other applications and purposes)? Many modules and Cmdlets in PowerShell can help you in general tasks like that. In the below, we use the ConvertTo-Csv Cmdlet with the Out-File parameter, which specifies the path of the CSV file.

Get-PowerBIWorkspace -Scope Organization -All | ConvertTo-Csv | Out-File c:\PowerBIWorkspaces.csv

The command above gives you a CSV file export of all workspace details as below;



Export all the Power BI workspaces to a CSV file

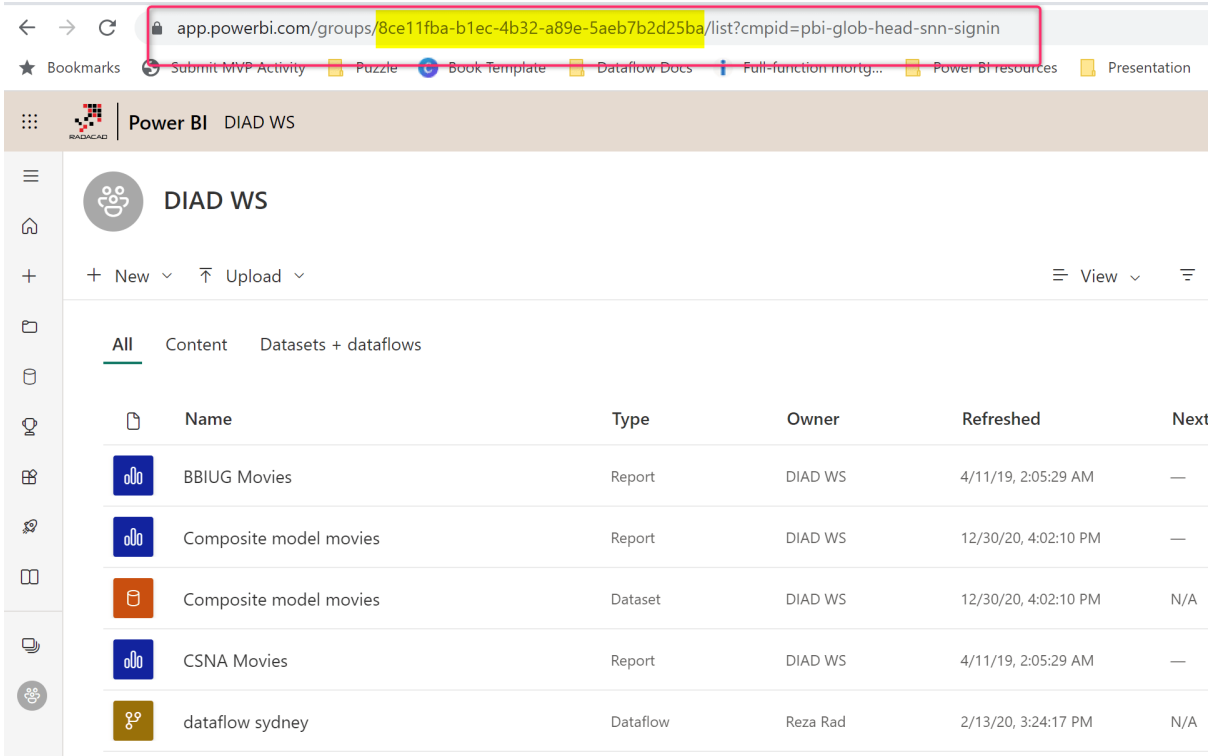
Isn't this helpful? Now imagine if you can schedule these simple few lines of script to run every night from a server and store list of workspaces. Now you have an application that stores the history of workspace changes throughout time in your organization. It is needless to say that you can generate the file name using Date and timestamp easily. So, if you are tasked to have a history of all workspaces throughout the entire organization at any certain point in time? No problem, you have the script above; schedule it. Here is an example of scheduling a PowerShell script using Task Scheduler.

Sample Scenario: Deploy a Power BI report from one workspace to another

PowerShell Cmdlets for Power BI can also create a deployment pipeline solution. This, then, can be a good replacement for the Deployment Pipeline feature, which is a Premium function in Power BI (of course, building something as comprehensive as that might take a while using the PowerShell Cmdlets).

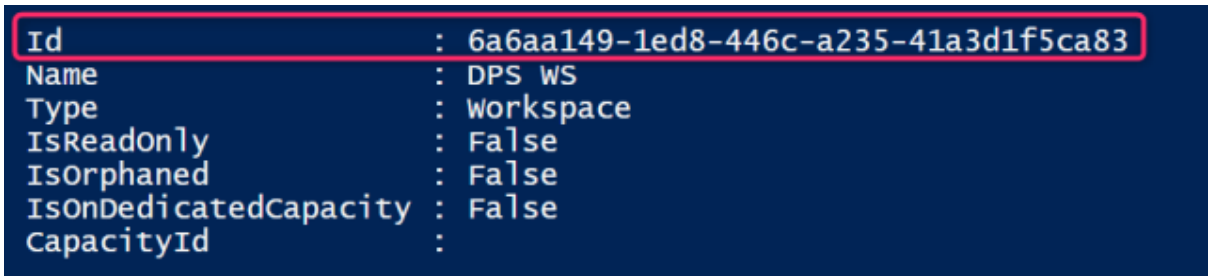
I am going to show you part of this as an example. Let's say we want to export a single report from one workspace to another. We can use a combination of Cmdlets for this purpose, including Get-PowerBIReport and Export-PowerBIReport.

11. You need to know the workspace Id to get a Power BI report from a workspace. The workspace Id is a unique identifier. This Id can be recognized in the page URL when you are inside the workspace.



Getting the workspace Id

The same workspace Id is also returned as the result of PowerShell Cmdlets such as Get-PowerBIWorkspace.



Getting the workspace Id from PowerShell Cmdlets

12. This workspace Id can now be used as a parameter for the Get-PowerBIReport Cmdlet to give you the list of all reports under the workspace.

Get-PowerBIReport -WorkspaceId <id of your workspace>

The result below shows the output.

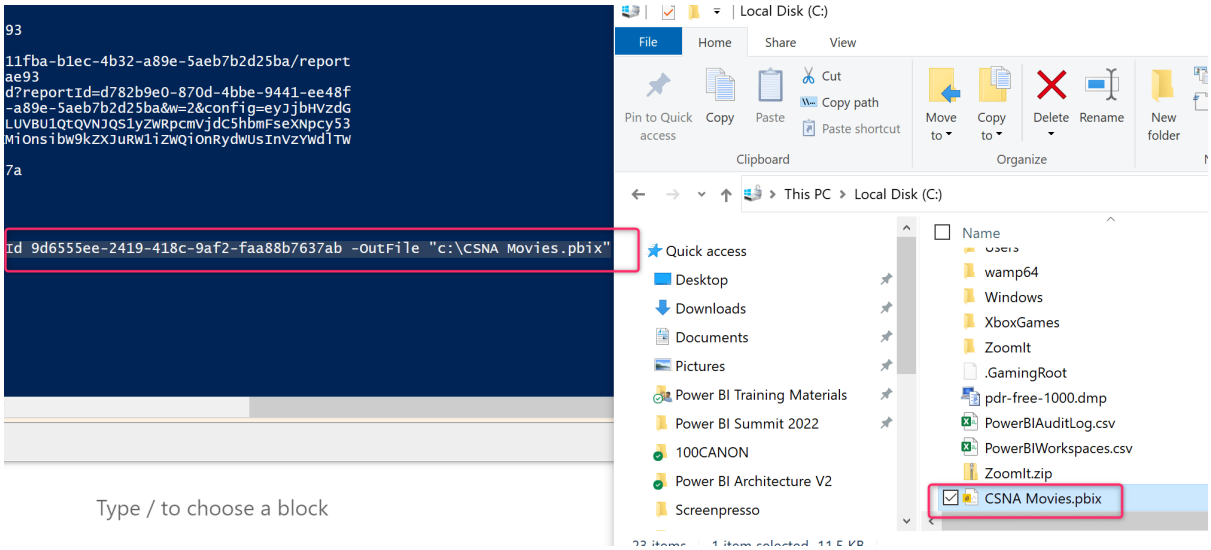
```
                vOcmljc1ZOZxhOIjpOcnVlfXO%3d
DatasetId : 74a44663-0659-4032-8907-0a5f66bad515

Id        : 9d6555ee-2419-418c-9af2-faa88b7637ab
Name      : CSNA Movies
WebUrl    : https://app.powerbi.com/groups/8ce11fba-b1ec-4b32-a89e-5aeb7b2d25ba/report
            s/9d6555ee-2419-418c-9af2-faa88b7637ab
EmbedUrl  : [illegible]

DatasetId : 39ef290c-6c77-4547-80f3-a53a92eb3156

Id        : d782b9e0-870d-4bbe-9441-ee48f313ae93
Name      : Composite model movies
WebUrl    : https://app.powerbi.com/groups/8ce11fba-b1ec-4b32-a89e-5aeb7b2d25ba/report
            s/d782b9e0-870d-4bbe-9441-ee48f313ae93
EmbedUrl  : [illegible]

DatasetId : d13dbba9-4f60-4efb-bb42-d20424be977a


PS C:\WINDOWS\system32>
```

Getting a list of Power BI reports under a workspace

13. Now to export the PBIX file of one of these reports, you can use the Report Id and the output file name like below;

Export-PowerBIReport -Id <report Id> -OutFile "c:\CSNA Movies.pbix"

This download the PBIX file for that report into the local path.

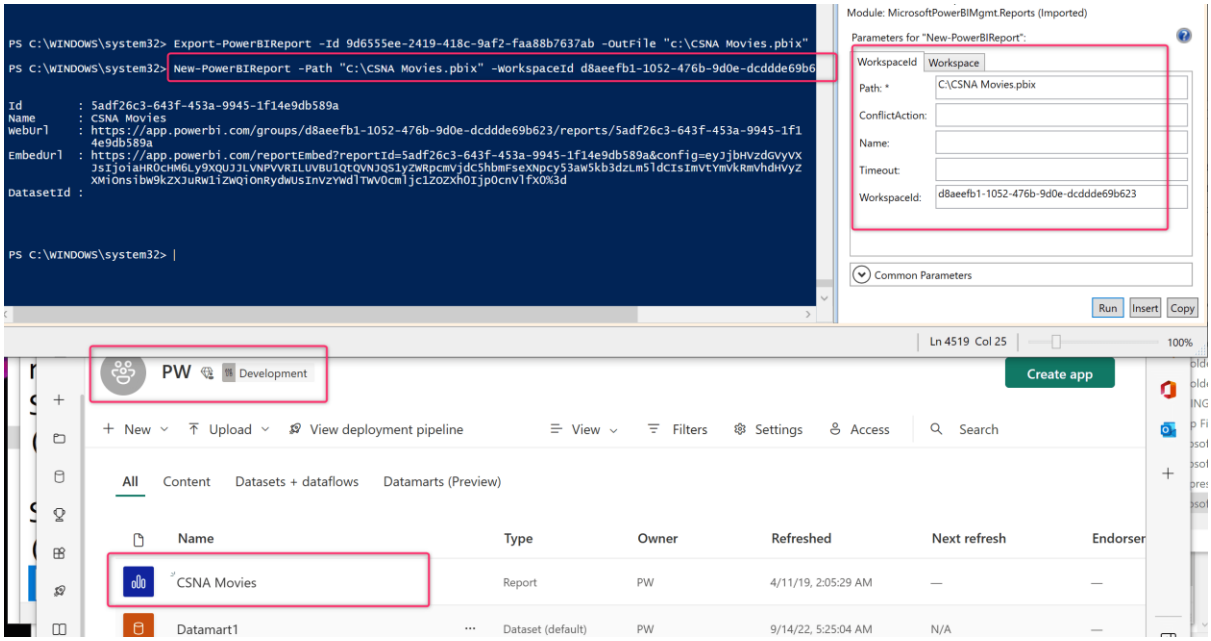


The PBIX file was exported using PowerShell Cmdlet.

14. Now we can publish this PBIX file to another workspace. You will need the workspace Id of the destination workspace, and you can use the Cmdlet below;

New-PowerBIReport -Path "C:\CSNA Movies.pbix" -WorkspaceId <workspace id>

The report is now copied into this destination workspace.



Publishing the Power BI report into a workspace

This action is different from the Copy-PowerBIReport Cmdlet. The difference between these two is that the Copy Cmdlet will create a live connection to the dataset in the source workspace, whereas the method I explained will create a copy of the dataset too.

The method above can be expanded into looping through all the reports in the source workspace and exporting them into the destination workspace, like a full deployment pipeline from the Development environment into the Test or Production environment.